uvt-disparity: Descriptor espacio-temporal para el reconocimiento de situaciones en entornos de tráfico

Trabajo Fin de Grado

MARIA DEL CARMEN RUDA FERNÁNDEZ

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
GRADO EN INGENIERÍA ELECTRÓNICA, INDUSTRIAL Y AUTOMÁTICA

Madrid, 2016

Documento maquetado con TEXIS v.1.0+.

uvt-disparity: Descriptor espacio-temporal para el reconocimiento de situaciones en entornos de tráfico

MARIA DEL CARMEN RUDA FERNÁNDEZ

Dirigida por el Doctor
BASAM MUSLEH LANCIS

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
GRADO EN INGENIERÍA ELECTRÓNICA, INDUSTRIAL Y AUTOMÁTICA

Madrid, 2016

Agradecimientos

En primer lugar, agradezco el simple hecho de poder poner por escrito en este apartado esos pensamientos que a menudo se tienen y tan pocas veces se expresan. Quiero agradecer ante todo a mi madre, por preocuparse tanto por mi, por sus visitas para darme apoyo y sus llamadas, por estar siempre ahí para darme ánimo; agradecer también a mi padre, por aguantar los agobios y cambios de humor y por cuidar tan bien de mi estos años. Gracias a mis hermanos, por entenderme y apoyarme a nuestra manera, por no necesitar de constantes llamadas para saber que siempre están ahí para mi, al igual que yo lo estoy para ellos. Gracias a mi familia en general, por interesarse aun en la distancia y darme todo su apoyo.

Gracias a mis amigos, por intentar buscar siempre un hueco para desconectar, por distraerme y darme tan buenos momentos. Gracias a él, a esa persona que apareció este año y se ha convertido en uno de los pilares mas fuertes en mi vida, por creer en mi incluso más de lo que creía yo misma, por no dejarme rendirme, por saber calmarme y ponerme las pilas al mismo tiempo, por sacarme de ese pesimismo tan mio y hacerme ver siempre el lado bueno de las cosas, por ser tu y por elegirme a mi.

Gracias a Basam, mi tutor de proyecto, por aconsejarme y guiarme tan bien desde el inicio y hasta el final de este trabajo, por estar siempre disponible para cualquier duda y por sus mensajes de apoyo; sin duda has hecho mucho mas de lo que te correspondía y no puedo si no agradecerte por ello.

Por último y aún a riesgo de parecer algo pelota, quería agradecer a mi jefe, por ser tan comprensivo y darme el tiempo que necesitaba para dedicarme a este proyecto, por dejarme salir antes y entrar después en cada ocasión en que lo pedí.

Gracias, en resumen, a todas aquellas personas sin las cuales habría sido mucho mas difícil, si no imposible, el escribir estas lineas; gracias de corazón.

Resumen

Hoy en día son numerosos los estudios sobre sistemas de ayuda a la conducción, tales como sistemas para la segmentación de los diferentes elementos de la escena, para la detección de obstáculos como vehículos y peatones, sistemas de navegación, etc; sin embargo, son menos los estudios dedicados a la detección de situaciones concretas en entornos de tráfico; siendo éste un factor decisivo en la toma de decisiones de cualquier conductor, así como es importante saber si un peatón se encuentra frente a nuestro vehículo, también lo es conocer que situación se presenta, pues un conductor no actúa de la misma forma frente al mencionado peatón si este se encuentra cruzando por delante del vehículo que si se encuentra parado frente al mismo.

En este trabajo se pretende crear un algorimo de visión por computador capaz de reconocer situaciones determinadas en entornos de tráfico a partir de la información captada por un par de cámaras en configuración estéreo situadas en la parte superior de un vehículo en movimiento. Para ello, una vez extraída la información tridimensional que proporciona el par de cámaras, se crea la imagen que se ha denominado para este trabajo como **uvt-disparity**, para generar dicha imagen, se parte del mapa de disparidad y la representación uv-disparity de una secuencia de imágenes tomadas consecutivamente para, de este modo, unir la información de los objetos u obstáculos que se encuentran en la imagen con su evolución a lo largo del tiempo. Este uvt-disparity posee por tanto información espacial y temporal del entorno en que se encuentra el vehículo.

Se propone como descriptor efectivo de este algoritmo el Histograma de Gradientes Orientados (HOG) para la creación de un clasificador mediante el método de aprendizaje supervisado de Máquinas de vectores de soporte (Support Vector Machines, SVM), junto con un estudio de la influencia de los parámetros del mismo para optimizar los resultados.

Índice

Índice de figuras

Índice de Tablas

Parte I

Memoria del proyecto

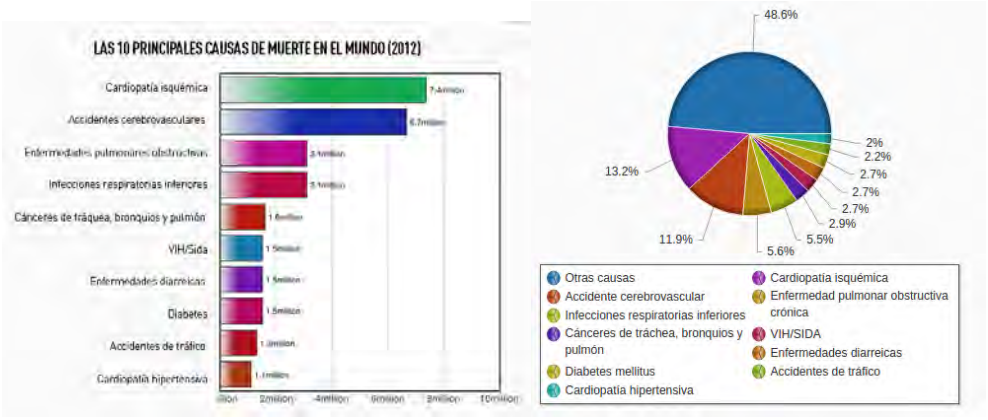
Capítulo 1

Introducción

1.1. Introducción y motivación

Hoy en día, los accidentes de tráfico continúan siendo una de las principales causas de mortalidad en el mundo (fig 1.1); según el Informe Mundial sobre Seguridad Vial de 2015 elaborado por la Organización Mundial de la Salud (OMS), fallecen 1.250.000 personas al año en el mundo debido a esta causa, y mas de 20 millones sufren traumatismos no mortales. Si bien es cierto que con el paso de los años, los accidentes mortales se han reducido drásticamente, siendo España el quinto país del mundo con mejor seguridad vial junto a Dinamarca y Reino Unido (fig 1.2); no por ello se debe dejar de investigar e invertir en métodos y sistemas para hacer el transporte por carretera, uno de los principales medios de desplazamiento globales, lo mas seguro posible tanto para vehículos como para peatones, ciclistas y motociclistas.

Debido a la gran demanda de esta mejora en seguridad vial, se ha generado un gran desarrollo de los sistemas de seguridad tanto en seguridad pasiva como en seguridad activa; es decir, con la finalidad de que, una vez producido el accidente, éste tenga el mínimo impacto posible sobre las personas, o con la finalidad de impedir o reducir estos accidentes. Actualmente podemos encontrar los dos tipos de sistemas en los vehículos comercializados, como el cinturón de seguridad, el áirbag, el ABS (Sistema Anti-bloqueo de ruedas), el sistema de frenada de emergencia, control de estabilidad, etc; y cada vez mas se incluyen otros métodos de seguridad activa mas novedosos como son los Sistemas Avanzados de Ayuda a la Conducción (ADAS, Advance Driver Assistant System), que emplean uno o varios sensores para captar información sobre el entorno del vehículo e informar al conductor o actuar en consecuencia si se considera necesario; como ejemplo de estos sistemas mas avanzados tenemos los sensores de aviso a la hora de aparcar e incluso los sistemas de frenado automático.



(a) (b)

Figura 1.1: Las 10 principales causas de defunción en el mundo (OMS, 2012)

Por último, el campo de la conducción autónoma también esta siendo profundamente investigado por sus grandes ventajas, como la optimización de la conducción, la eliminación de accidentes por distracciones del conductor y la posibilidad de gestionar mas eficientemente el tiempo al volante para el pasajero.

1.2. Entorno socio-económico y marco regulardor

Los accidentes de tráfico conllevan una serie de costes que justifican completamente la inversión en métodos de seguridad para evitar los mismos; son los costes administrativos, materiales y los mas importantes, los asociados a las víctimas, estos últimos, para su cuantificación pueden dividirse en costes humanos, pérdidas de producción por bajas y costes médicos (fig 1.3).

En España, el coste social asociado a las víctimas de accidentes de tráfico representa aproximadamente el 2 % de todo el producto interior bruto; siendo equivalente a prácticamente un tercio de la riqueza que genera en España todo el sector automovilístico (Fitsa, 2008). Por todo esto, tanto el gobierno como los propios fabricantes de vehículos invierten profundamente en el tema de seguridad, fijándose tanto en el bienestar de los pasajeros del automóvil como de los peatones y vehículos de su alrededor.

Existen tres factores que interviene en toda situación de tráfico: el conductor, el vehículo y el entorno; y una gran parte de los accidentes de tráfico son debidos principalmente al primer factor, el factor humano, a causa de

Figura 1.2: Evolución de los accidentes mortales en vías interurbanas desde 1960 (OMS, 2016)

distracciones o debido a la conducción bajo efectos físicos y psicológicos no adecuados. Uno de los parámetros decisivos a la hora de evitar un accidente es el tiempo de reacción del conductor ante determinada situación (fig 1.4); éste es otro de los motivos por los cuales se está invirtiendo tanto en los sistemas de ayuda a la conducción, que pueden dotar al conductor de información sobre obstáculos o situaciones que puede no haber percibido aún o incluso actuar si la situación lo exige.

1.3. Objetivos

Como ya se ha visto, con el paso de los años, la seguridad en los vehículos actuales ha aumentado considerablemente, reduciéndose el número de accidentes de tráfico generados por los mismos. El interés en este factor se plasma en programas de mejora y comparación de sistemas de seguridad, como lo es el conocido Programa Europeo de Evaluación de Automóviles Nuevos (NCAP, x), que informa a conductores y compradores de las mejoras en seguridad y en sistemas de ayuda a la conducción, premiando e incentivando la evolución de los mismos.

El objetivo del presente trabajo es la creación de un sistema de ayuda a la conducción basado en técnicas de visión por computador para el reconocimiento de situaciones determinadas en entornos de tráfico tanto en zonas urbanas como interurbanas capaz, no solo de detectar los objetos presentes en la escena sino de describir la acción que realizan los mismos. Para

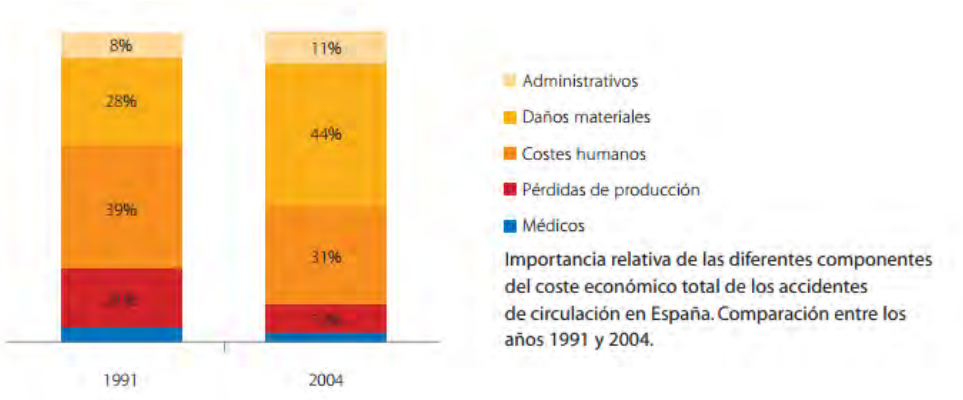


Figura 1.3: Distribución de los costes asociados a los accidentes de tráfico (Fitsa, 2008)

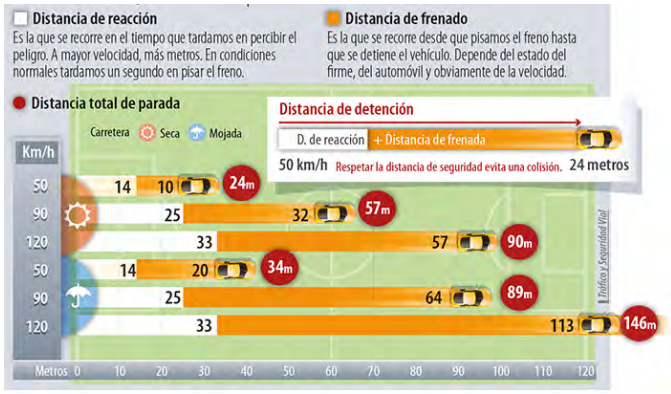


Figura 1.4: Distancias recorrida por el vehículo en los tiempos de reacción del conductor y frenado del mismo en función de la velocidad de circulación (DGT, 2015)

ello, se emplea un sistema de cámaras estéreo situadas en la parte superior del vehículo que se pretende monitorizar y se estudian secuencias de cuatro imágenes captadas por el sistema en movimiento en distintas situaciones y entornos. El empleo de este sistema se justifica por la gran variedad de objetos y texturas que se pueden encontrar en las múltiples situaciones de tráfico posibles, cuya comprensión requiere de una gran cantidad de información, como es la proporcionada por una imagen; además, la capacidad de obtener información tridimensional del par estéreo permite segmentar fácilmente los elementos correspondientes a obstáculos u objetos en la imagen de los correspondientes a la calzada y fondo de la carretera.

El algoritmo implementado puede ser empleado como complemento para diferentes sistemas ADAS con el fin de aumentar la información que estos

proveen, incrementando así su eficacia y mejorando la monitorización del vehículo.

1.4. Organización del documento

El presente documento se encuentra organizado de la siguiente manera:

- El primer capítulo, donde se ha repasado la motivación, razón y objetivos del proyecto, así como su entorno socio-económico.
- El capítulo dos, donde se realiza una síntesis del estado del arte actual y de las investigaciones y estudios realizados en la materia que atañe al proyecto a lo largo de los últimos años. Se profundiza también en este capítulo en los estudios que han servido como base a este trabajo, explicando sus algoritmos y conceptos fundamentales.
- En el capítulo tres, dedicado a la implementación práctica, ahonda en los algoritmos y métodos empleados, añadiendo información visual de resultados, comparaciones y métodos seguidos, así como los tiempos de cómputo de cada paso.
- En el capítulo cuatro, dedicado también al estudio de resultados, se diferencian los datos obtenidos en función de la situación concreta a clasificar de un modo mas detallado.
- En el último capítulo se expone la conclusión final sobre el estudio realizado y las posibles mejoras y trabajos futuros.
- La bibliografía mencionada a lo largo del trabajo, se recoge en su apartado correspondiente, en la parte final del documento.
- Por último, se anexa el apartado sobre la planificación y tareas seguidas para la realización del proyecto, así como el presupuesto y tiempo empleado en el mismo.

Capítulo 2

Estado del arte y conceptos

Este capítulo esta dedicado a la recopilación del estado actual de desarrollo de los diferentes métodos, soluciones y líneas de investigación para la ayuda y automatización de la conducción mediante algoritmos de aprendizaje y visión por computador, centrándose mas específicamente en la detección de situaciones y acciones en determinados entornos.

Se explicarán mas detalladamente aquellos estudios y conceptos que estén directamente relacionados con la metodología que se ha seguido en este proyecto, o que definan alguno de los conceptos básicos para la comprensión del mismo.

2.1. Introducción: sistemas de visión por computador

La visión por computador puede definirse como la disciplina que, mediante diversos métodos, pretende adquirir, analizar y procesar las imágenes del mundo real con el fin de obtener de ellas información numérica o simbólica para que pueda ser tratada por un computador.

En un sistema de visión por computador es imprescindible, por tanto, la adquisición de imágenes para su posterior tratamiento; estas imágenes suelen adquirirse mediante sistemas de cámaras monoculares o estéreo. Mientras que en los sistemas monoculares se dispone de una única cámara, en un sistema estéreo disponemos de dos cámaras que permiten obtener dos imágenes de un mismo instante desde diferentes puntos de vista; esta configuración posibilita la obtención de información tridimensional de dicho instante. En el caso de un sistema monocular se puede obtener también cierto grado de información tridimensional siempre que la cámara no se encuentre estática y realice un movimiento conocido.

Los sistemas monoculares se utilizan habitualmente en sistemas de navegación (Royer et al., 2007) sobre bases móviles y en mapas previamente entrenados ,en la detección y clasificación de obstáculos como vehículos (Ponsa et al., 2005) y personas (Enzweiler y Gavrila, 2009) y en la segmentación de escenarios (Alvarez et al., 2012) principalmente. Los sistemas de este tipo emplean diferentes algoritmos sobre la imagen basados en información sobre iluminación, color, la geometría del objeto que se pretende detectar.

Respecto a los sistemas estéreo, si bien la información 3D que aportan es una ventaja frente a los sistemas monoculares, esta misma información provoca que los algoritmos aplicados deban procesar una mayor cantidad de datos. En este ámbito existen numerosos estudios para su aplicación de nuevo en sistemas de navegación (Murray y Little, 2000) , detección de obstáculos y segmentación de escenarios (Nedevschi et al., 2004) . El trabajo propuesto se basa en un sistema de cámaras estéreo, por lo que se va a profundizar mas en este tipo de sensores.

Es frecuente también la incorporación de otros tipos de sensores que completen e incrementen la información que las cámaras no son capaces de detectar para optimizar y aumentar la precisión del sistema, como pueden ser sensores láser o de proximidad.

2.1.1. Sistema estéreo, extracción de la información

Como se ha mencionado anteriormente, podemos obtener información tridimensional del entorno a partir del sistema de cámaras estéreo, este sistema consiste en tomar dos imágenes en el mismo instante de tiempo desde diferentes puntos de vista, para ello se emplean dos cámaras (cámara izquierda y derecha) situadas a la misma altura respecto del suelo, con una distancia entre ellas determinada B, con sus ejes ópticos paralelos O y con una distancia focal idéntica f (fig 2.1).

Con el fin de obtener la información tridimensional, se conoce que la profundidad espacial a la que se encuentran los puntos de cada uno de los píxeles de las imágenes del par estéreo, es proporcional a la diferencia en coordenadas de la imagen en la que se proyecta dicho punto en cada uno de los planos de la imagen, así, aquellos objetos que presenten un desplazamiento mayor en una imagen respecto de la otra, se encontrarán a una profundidad menor y viceversa; esta diferencia es lo que se conoce como disparidad y se calcula resolviendo el llamado problema de correspondencia, que se ve simplificado si las imágenes del sistema estéreo están previamente rectificadas.

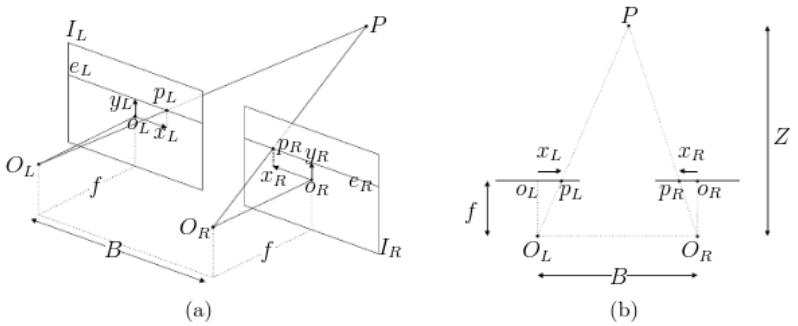


Figura 2.1: Configuración de las cámaras del par estéreo y relación de parámetros (Lecumberry, 2005)

Existen dos tipos de mapas de disparidad, el disperso, en el que solo se recopila la información de los bordes de los elementos de la imagen y el utilizado en este proyecto, el mapa de disparidad denso, que refleja los valores de disparidad para todos los píxeles de la imagen (fig 2.2). Este mapa consiste en una imagen en la que el nivel de gris de los píxeles corresponde al valor de disparidad (proporcional a la distancia del punto en el mundo) del par de píxeles del sistema estéreo. Los puntos mas alejados tendrán un valor de disparidad menor, por lo que en el mapa aparecerán con niveles de gris próximos al negro y viceversa.

Figura 2.2: Imagen original y mapas de disparidad disperso y denso (Llorca et al., 2012)

Para resolver este problema de correspondencia, existen numerosos estudios basados en 3 pasos principales (Scharstein y Szeliski, 2002):

1. Cálculo de la función de coste: cálculo de las diferencias entre píxeles en el par de imágenes.

2. Cálculo del coste de agregación: agregación de los distintos costes calculados por grupos o ventanas de píxeles.

3. Cálculo del valor de disparidad normalmente en función del coste de

agregación.

Siendo el algoritmo de H.Hirschmuller (Hirschmu, 2008) el escogido para este proyecto por su mayor velocidad de procesamiento que la mayoría de otros métodos. Éste algoritmo presenta un método de correspondencia estéreo (Semiglobal Matching - SGM) que emplea un procesamiento de píxeles local basado en comparar el coste de distintas correcciones radiométricas (variaciones de intensidad del píxel) de las imágenes de entrada, de este modo, el método SGM permite una rápida aproximación por optimizaciones para todas las direcciones.

En el algoritmo de H.Hirschmuller se calcula la diferencia de intensidades entre píxeles correspondientes; la función de coste, por "Información Mutua" y el valor de agregación del coste como una aproximación de la función de energía por optimizaciones en todas las direcciones de la imagen. La disparidad se calcula finalmente mediante el método "Winner takes all", en el que se selecciona la disparidad correspondiente al menor coste, y con ajustes como comprobación de consistencia e interpolación de píxeles.

El propio artículo propone como alternativa al uso de la Información mutua, que se define como la entropía de ambas imágenes calculada a partir de la función de probabilidad de las intensidades y cuyo coste computacional es muy elevado; emplear el método simplificado de Birchfield-Tomasi (Birchfield y Tomasi, 1998), donde la función de coste se calcula como la mínima diferencia de intensidades entre píxeles correspondientes en la media del píxel en cada dirección a lo largo de la línea epipolar. El cálculo de la función de coste puede ser ambiguo debido al ruido de la imagen, para reducir esto, se añaden constricciones adicionales como la penalización en el cambio de disparidad de píxeles vecinos.

Una vez construido el mapa de disparidad es frecuente la implementación de las imágenes u-disparity y v-disparity a partir del mismo (Labayrade et al., 2002) y (Hu y Uchimura, 2005) . Estas imágenes contienen una acumulación de los histogramas laterales del mapa de disparidad bien por columnas o por filas respectivamente, permitiendo obtener una representación del contenido geométrico de la imagen sin influencias de iluminación. En la imagen resultante del v-disparity puede verse, por tanto, información de la geometría de la calzada como una línea oblicua así como de los obstáculos presentes representados como lineas verticales proporcionales a su dimensión y de valor de intensidad correspondiente a su disparidad; mientras que en la del u-disparity se obtiene una representación de la distribución de estos mismos obstáculos frente al vehículo con un valor proporcional a su altura medida en píxeles (fig 2.3).

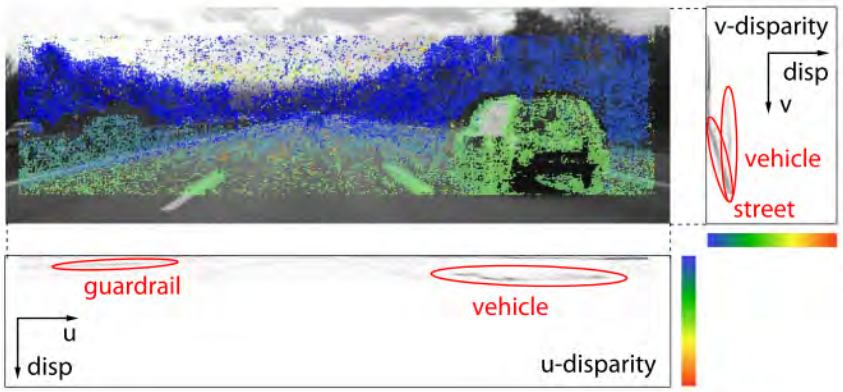


Figura 2.3: Ejemplo mapa de disparidad e imágenes uv-disparity (Teutsch et al., 2010)

Por último, la representación de la imagen imagen uv-disparity se ha utilizado ampliamente en la segmentación de escenarios entre obstáculos y espacio libre, bien por sí misma (Perrollaz et al., 2010) o con información adicional de otros sensores para mejorar el rendimiento en espacios mas complejos (Teutsch et al., 2010). El espacio libre suele obtenerse a partir de la información del perfil de calzada proporcionada por el v-disparity, mientras que en la detección de obstáculos el algoritmo se suele centrar en la determinación de regiones de interés representativas que permitan reducir la información a procesar.

2.2. Reconocimiento de la escena, detección de acciones y situaciones

Muchos de los estudios sobre sistemas de ayuda a la conducción mediante visión por computador se basan principalmente en la segmentación y reconocimiento de la escena frente al vehículo y la clasificación de los elementos que en ella se encuentran. Algunos métodos separan primero los los tres planos propios de una situación de tráfico: cielo, fondo y carretera para después centrar la clasificación de objetos en este último (Ess et al., 2009) y (Geiger et al., 2014) (fig 2.4). Otros emplean información sobre el color e imágenes infrarrojo para reconocer la configuración espacial de objetos y extraer el contexto de la imagen (Kang et al., 2011). Sin embargo, este apartado se centrará en aquellos métodos que buscan la detección de acciones y situaciones concretas en un contexto determinado.

Sobre la detección de acciones se han realizado algunos estudios princi-

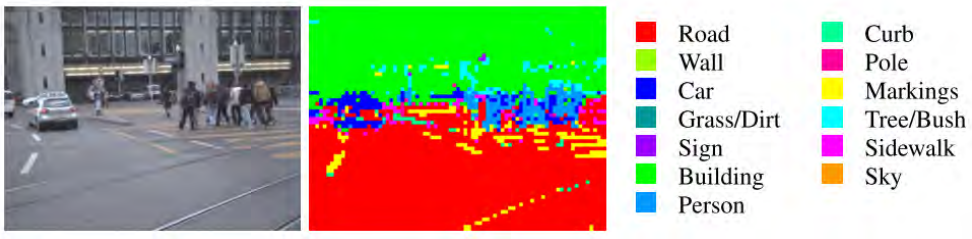


Figura 2.4: Reconocimiento de la escena por segmentación de elementos (Ess et al., 2009)

palmente centrados en la detección de acciones en personas; las aplicaciones típicas para esta detección incluyen vídeo-vigilancia, reconocimiento de gestos o análisis de eventos deportivos, normalmente a partir de secuencias de vídeo. Se han propuesto para este fin numerosos detectores y descriptores (Wang et al., 2009), centrándose en entender y describir las acciones humanas como patrones de movimiento de píxeles y regiones de interés.

El estado del arte actual sobre los métodos de clasificación de acciones incluyen características globales espacio-temporales a partir de una secuencia de vídeo. No obstante, estas características pueden ser en parte debidas a partes irrelevantes del contexto de la escena; esto ha motivado que algunos estudios se centren en partes o regiones concretas de la imagen deformables en el espacio y el tiempo y en características invariantes a los cambios de localización. Se transforma por tanto la información en vídeo a una representación en bolsa de palabras (bag-of-words, BoW) (Wallach, 2006) , de características (bag-of-features, BoF) (Lazebnik y Schmid, 2006) o en vectores Fisher (Oneata et al., 2013) ; para extraer los puntos de interés se recurre frecuentemente al operador Harris, capaz de detectar las esquinas de los objetos en la imagen. A partir de los puntos de interés, las regiones representativas detectadas corresponden a extremos de movimiento y características basadas en dinámicas de gradiente y flujo óptico. En (Sapienza et al., 2014) emplea como descriptores el Histograma de Gradientes Orientados (Histograms of Oriented Gradients, HoG) (Dalal y Triggs, 2005) y el Histograma de movimiento de contornos (Motion Boundary Histogram, MBH) (Navneet Dalal y Schmid, 2006).

En (Marszalek Marcin. Laptev Ivan, 2009) emplea como descriptores el Histograma de Gradientes Orientados (HoG), el Histograma de flujo óptico (HoF) y el descriptor SIFT (Scale-invariant feature transform); en este caso la unión de los tres descriptores y la mezcla de información bidimensional y tridimensional permite capturar patrones de movimiento a través del histograma de flujo óptico, apariencia dinámica sobre la información tri-

dimensional gracias al histograma de gradientes orientados y la apariencia estática sobre la bidimensional mediante el SIFT.

Como puede observarse, son numerosos los estudios que emplean el Histograma de Gradientes Orientados (HoG) como descriptor fundamental espacio-temporal (Klaser et al., 2008) añadiéndole información de otros descriptores para obtener todos los datos necesarios y optimizar el proceso. La utilización de descriptores de Histograma de Gradientes Orientados, introducido en (Dalal y Triggs, 2005) para la detección de personas, se basa en la evaluación de histogramas locales normalizados de gradientes orientados en regiones densas. La idea básica es que la apariencia local y tamaño de un objeto pueden caracterizarse por la distribución local de sus gradientes de intensidad o dirección de sus bordes, incluso sin la necesidad de un cocimiento preciso de los gradientes correspondientes o la localización de dichos bordes. El mencionado algoritmo, supuso una gran contribución a los sistemas de visión por computador por sus buenos resultados y no muy compleja implementación; por todo ello, el descriptor HOG es el escogido para este trabajo, describiéndose detalladamente el método y su algoritmia mas adelante en el capítulo sobre la implementación práctica (sección 3.4).

En cuanto a la detección de acciones y situaciones en entornos de tráfico existen un menor número de artículos, destacando aquellos que se basan en detectar el vehículo en cuestión y seguir su movimiento a través de la secuencia de imágenes para, a través de un árbol de decisiones ir acumulando las situaciones y así poder predecir e identificar la siguiente. Al estar prefijado el árbol de situaciones, lo hace complejo de implementar e incapaz de aprender nuevas acciones del vehículo, pues habría que añadirlas; por este motivo, si bien puede resultar útil para la monitorización del tráfico de una zona concreta, donde todas las acciones posibles son fácilmente predecibles como ocurre en (Haag y Nagel, 2000) , no resulta un método óptimo para la asistencia al conductor. Otros como (Graefe, 1992) , utilizan una mayor cantidad de cámaras y sensores sobre el vehículo base para detectar lo que ocurre en el entorno del vehículo (Reconocimiento del carriles y de las líneas marcadas entre ellos, reconocimiento de otros vehículos, obstáculos estáticos y señales) de este modo y con toda esa información se puede conocer la situación y actuar acorde a ella, sin embargo este método está pensado para autopista, donde los coches detectados serán móviles y no habrá un exceso de obstáculos ni coches aparcados. Los módulos de reconocimiento para cada situación deben comprobarse por separado e independientemente; si se quisiera detectar alguna otra habría que añadir otro módulo.

Los sistemas mas actuales, entendiendo una situación como una distribución de motivaciones, metas y planes (Dagli et al., 2002) o simplemente

como secuencias de estados (Meyer-Delius et al., 2009) con algún significado, emplean Redes de Conocimineto Dinámicas (Dynamic Belief Networks, DBN) como el Modelo Hidden Markov (HMM) para describir estas distribuciones y generar las matrices de transición con las probabilidades de pasar de un estado a otro, el cual requiere de un entrenamiento previo en el que es importante el número de estados que definen la situación, pues si los estados son pocos, el resultado será muy general.

2.3. Sistemas de entrenamiento

Finalmente, para la clasificación y diferenciación de las clases que se pretenden detectar, es cada vez mas común el uso de Métodos de Aprendizaje (Machine Learning) que permiten crear la función que relaciona los parámetros del descriptor de entrada con las salidas de interés o clases.

En este ámbito existen dos tipos de aprendizaje, el supervisado y el no supervisado; como indica el propio nombre, en el aprendizaje no supervisado no es necesario un conocimiento a priori de la clasificación de las imágenes de entrada, sino que el modelo se va ajustando automáticamente, mientras que en el aprendizaje supervisado se requiere de un entrenamiento previo del sistema con ejemplos de las posibles situaciones a clasificar debida y correctamente etiquetadas. Por la mayor simplicidad y adaptabilidad del método, para este trabajo se emplea el aprendizaje supervisado; el inconveniente de este método radica en que para obtener resultados muy precisos, se requiere una gran cantidad de ejemplos de situaciones diversas debidamente etiquetadas para el entrenamiento, por lo que el número de imágenes ejemplo del que se disponga será un factor decisivo en los resultados, como se verá mas adelante. Dentro del aprendizaje supervisado existen varios métodos, desde el k-vecino-mas-próximo (k Nearest Neighbors classifier), que es el mas sencillo, donde simplemente se comprueba la distancia del nuevo caso a clasificar con las de los casos ya existentes, escogiendo los k casos mas similares; hasta las mas complejas, como son las Redes Neuronales o las Máquinas de Soporte Vectorial (Support Vector Machines, SVM).

- **Redes Neuronales**

  Nacidas como una simulación de los sistemas neuronales biológicos, las redes neuronales utilizan algoritmos de aprendizaje adaptativo y autoorganización con funciones no-lineales y procesamiento en paralelo, las neuronas pertenecientes a la red reciben una serie de información de la entrada que comprueban y contrastan con los datos del entrenamiento

previo a través de interconexiones para proporcionar una de las salidas definidas; cada neurona tiene asociada una función matemática denominada función de transferencia, que es la relación entre la señal de entrada y la de salida del sistema, a partir de la cual, unida al peso de las conexiones entre neuronas, permiten obtener la clasificación.

- **Support Vector Machines**

Tomando los datos de entrenamiento como conjuntos de vectores de clasificación en un espacio n-dimensional, el método de aprendizaje supervisado de Máquinas de Soporte Vectorial tiene como objetivo encontrar el hiper-plano que separe y diferencie las distintas clases, buscando además aquel con mayor margen entre ellas (fig 2.5). Este hiper-plano de separación no tiene por que ser lineal, sino que puede tomar varias formas gracias a la implementación de funciones Kernel, que permiten proyectar la información en espacios de características de mayores dimensiones, siendo las mas conocidas la funcion RBG (Base Radial Gausiana), la polinómica o la tangencial.

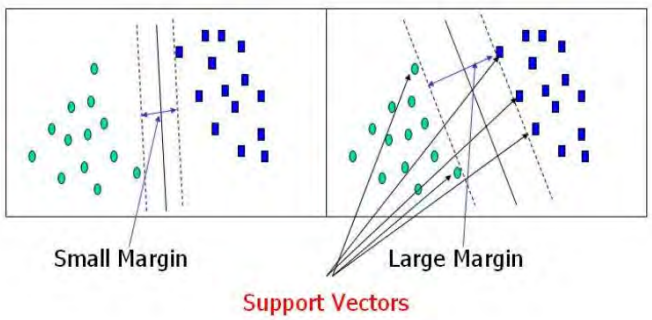


Figura 2.5: Metodología de separación de clases en SVM (Montes, 2015)

Por la adaptabilidad de este método de aprendizaje a las distintas distribuciones de datos, este será el sistema empleado en el presente trabajo, explicándose más detalladamente en el apartado sobre implementación práctica (sección 3.6)

2.3.1. Evaluación de la calidad del clasificador

Para escoger el clasificador mas óptimo es necesario recurrir a métodos de evaluación del mismo, que proporcionan indicadores de calidad y precisón; para ello existen distintas metodologías, representaciones gráficas y herra-

mientas, que pueden tomar tres enfoques diferentes:

- **Puntuación del estimador**: los propios estimadores tienen un método de puntuación que proporciona un criterio de evaluación por defecto y que proporciona un porcentaje de precisión y de error para el problema que están destinados a resolver.

- **Puntuación de los parámetros**: las herramientas de evaluación de modelos utilizan validación cruzada para estudiar la influencia de los distintos parámetros del clasificador. Esto consiste en realizar el entrenamiento repetidas veces con diferentes particiones de datos de entrenamiento y prueba para cada parámetro influyente, evitando así el sobreajuste de estos parámetros para un único caso concreto de entrenamiento y obteniendo de esta forma resultados fiables y comparables.

- **Funciones métricas de clasificación**: miden los resultados de la clasificación, evaluando la precisión y errores del mismo y pudiendo ser representadas en una gráfica para una interpretación visual de los resultados, las empleadas en este proyecto en concreto son las curvas *precision-recall* y las curvas *ROC* (Receiver operating characteristic).

  -Precision-recall

  Respecto a la clasificación final obtenida, la precisión es la medida de la relevancia de los resultados, es la capacidad de sistema para relacionar cada caso individual con su clase correspondiente; mientras que el *recall* es la medida de cuantos resultados verdaderamente relevantes son devueltos, es decir, la capacidad para identificar todos los elementos de esta clase. El mejor resultado será aquel que presente una mayor área bajo la curva, lo que representa un alto nivel de estos dos parámetros: una alta precisión significará un bajo nivel de falsos positivos y un alto *recall*, un bajo nivel de falsos negativos.

  La precisión se define como el numero de verdaderos positivos entre el numero total de positivos detectados.

$$precision = \frac{tp}{tp + fp}$$

El *recall* se define como el numero de verdaderos positivos entre el numero total de positivos reales.

$$recall = \frac{tp}{tp + fn}$$

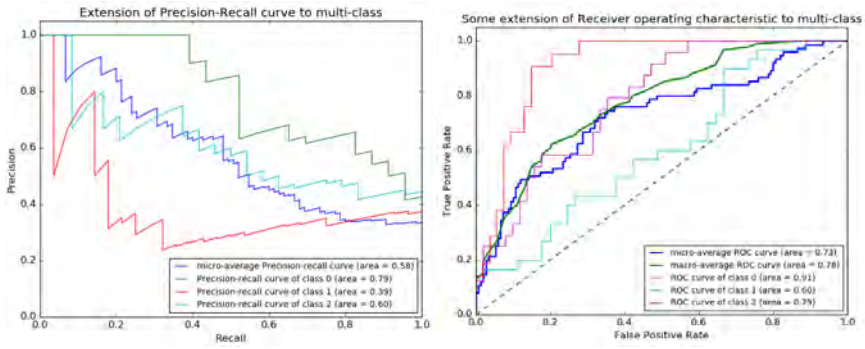
Asi, un sistema con un alto *recall* pero una baja precisión retornará muchos resultados positivos, pero muchos de ellos serán incorrectos. De forma inversa, un sistema con una alta precisión pero un bajo *recall* devolverá pocos resultados, pero la mayoría de ellos serán correctos. Por este motivo, un sistema ideal será aquel con un valor alto de ambos parámetros.

-ROC (Receiver Operating Characteristic)

Las curvas ROC representan el ratio de verdaderos positivos frente al de falsos positivos según varía el umbral de discriminación o valor a partir del cual se considera un caso como positivo; de este modo si disminuye el valor de correlación, disminuyen los falsos negativos elevando los positivos y viceversa. La clasificación será mas efectiva cuando haya un gran número de verdaderos positivos en relación a una baja cantidad de falsos positivos.

A partir de estas curvas y con el fin de simplificar su resultado a un único valor numérico, se utiliza también como indicador de rendimiento del clasificador su área interna, el Área Bajo la Curva (AUC, Área Under the Curve), que puede interpretarse como la probabilidad de que ante dos casos de cada tipo, el sistema clasifique a cada uno de ellos correctamente.

Se emplearán para este proyecto los métodos de comprobación por validación cruzada y las representaciones visuales de las curvas Precision-recall y ROC



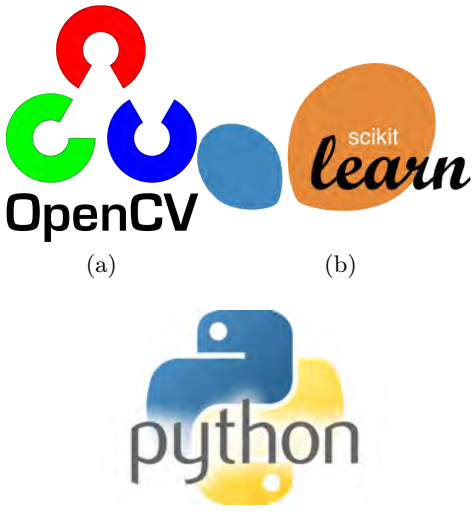
(a) (b)

Figura 2.6: Ejemplo curvas de calidad Precision-Recall (a) y ROC(b), visualmente se comprueba que las clases con una clasificación mas óptima son la verde y la roja respectivamente, pues poseen un mayor área bajo la curva (Pedregosa et al., 2011)

Capítulo 3

Implementación práctica

Para el desarrollo de la solución propuesta se han implementado distintos programas en lenguaje de programación Python y empleando librerías de visión por computador OpenCV (Bradski, 2000) y Sckit-image (van der Walt et al., 2014) para el tratamiento de imágenes y Scikit-learn (Pedregosa et al., 2011) para los algoritmos de entrenamiento y clasificación (fig 3.1).



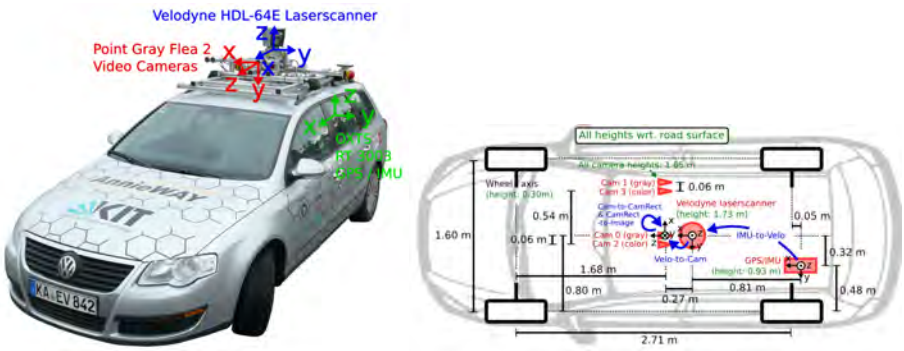
(a) (b)

(c)

Figura 3.1: Librerías utilizadas para el tratamiento de imágenes (a), funciones aprendiaje SVM (b) y lenguaje de programación empleado (c)

En este capítulo se detallan todas las tareas implementadas para el sistema, los algoritmos empleados, sus parámetros críticos y la influencia y resultados de los mismos.

3.1. Adquisición de la base de datos de imágenes



(a) (b)

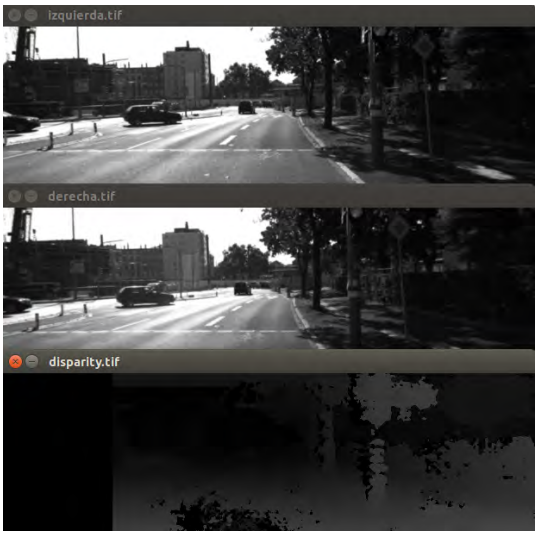
Figura 3.2: Sistema estéreo empleado, cámaras Point Gray Flea 2 Video Cameras en (a) y Cam 3 y Cam 2 en (b), ((KITTI, 2016))

Como se ha mencionado anteriormente, el sistema propuesto se basa en un sistema de cámaras estéreo, la base de datos escogida para esta implementación ha sido la base de datos de acceso público perteneciente al proyecto del Instituto de Tecnología Karlsruhe y al Instituto Tecnológico Toyota en Chicago (Geiger et al., 2012). Como se puede ver en la imagen 3.2, el sistema estéreo empleado para generar la base de datos utilizada consta de dos cámaras situadas en la parte superior del vehículo, éstas proporcionan parejas de imágenes estéreo en una secuencia de hasta 4 instantes consecutivos, con una diferencia de tiempo fija entre imágenes; en estas se pueden ver diversas situaciones típicas de entornos de tráfico: peatones, ciclistas, vehículos y trenes, bien cruzando frente al vehículo o circulando en paralelo a él; intersecciones, pasos de cebra, entornos interurbanos sin obstáculos, entornos urbanos con multitud de elementos, etc. Esta diversidad de casos en la base de datos permite un entrenamiento del sistema mas completo, pues se podrían tener en cuenta todas las situaciones incluidas.

Es importante mencionar que las imágenes de esta base de datos pública se encuentran ya debidamente rectificadas, factor que, como ya se mencionó en el apartado sobre el estado del arte (capítulo 2) permite la simplificación del problema de correspondencia en la creación del mapa de disparidad; partiendo de ello, la rectificación no será una tarea a realizar por el sistema propuesto.

3.2. Obtención del mapa de disparidad

(a) (b)



(c)

Figura 3.3: Resultados de los mapas de disparidad a partir de las imágenes izquierda y derecha en los casos de estudio, un vehículo delante (a), un peatón cruzando (b) y calzada libre (c)

A partir de las dos imágenes de entrada y con el fin de extraer la información tridimensional del sistema estéreo, se calcula el mapa de disparidad, introducido anteriormente en la sección 2.1.1 del capítulo sobre el estado del

arte, que se traduce como el índice de correspondencia entre los píxeles del par de imágenes de las cámaras izquierda y derecha. Cuanto mas cercano se encuentre un objeto, habrá un mayor desplazamiento del mismo en una imagen respecto de la otra, por lo que su disparidad será mayor; de la misma forma, un objeto alejado tendrá una disparidad mas baja. Para este cálculo, se ha utilizado una función de las librerías Opencv basada en el algoritmo de H.Hirschmuller (Hirschmu, 2008), también definido en la sección citada anteriormente, con las siguientes modificaciones:

- Considera 5 direcciones en lugar de las 8 consideradas originalmente.

- Busca coincidencias por grupos de píxeles llamados bloques.

- Utiliza el simplificador Birchfield-Tomasi para la función de coste (Birchfield y Tomasi, 1998)

- Incluye filtros pre y post procesamiento

El algoritmo, basándose en el estudio citado e implementando las modificaciones mencionadas, sigue los pasos definidos a continuación:

1. **Cálculo de la función de coste**:

   La función de coste se calcula como la mínima diferencia de intensidades entre grupos de píxeles correspondientes en la media del píxel en cada una de las cinco direcciones consideradas y a lo largo de la linea epipolar.

   Asumiendo que la imagen de entrada tiene una geometría epipolar conocida y esta debidamente rectificada, se obtiene un array uni-dimensional de valores de intensidad I_L y I_R mediante el muestreo de las funciones de intensidad uni-dimensionales resultado de la convolución de la incidencia de la luz en las imágenes izquierda y derecha respectivamente. La finalidad es obtener la disparidad entre un píxel en la posición x_L de la imagen izquierda y su correspondiente par x_R en la imagen derecha. Siendo I'_R la función de interpolación lineal entre los puntos de la imagen derecha, se mide como de bien encaja la intensidad de x_L en la región de interpolación lineal alrededor de x_L. Se obtienen las siguientes expresiones:

$$d'(x_L, x_R, I_L, I_R) = min \left| I_L(x_L) - I'_R(x) \right|$$

$$d'(x_R, x_L, I_R, I_L) = min \left| I_L(x) - I'_R(x_R) \right|$$

Para los intervalos $(x_R - 1/2 \leq x \leq x_R + 1/2)$ y $(x_L - 1/2 \leq x \leq x_L + 1/2)$ respectivamente.

Finalmente, el coste $C(x_R, x_L)$ se calcula simétricamente como el valor absoluto de la mínima diferencia de intensidades de ambas imágenes: entre píxeles se define simétricamente como el mínimo valor entre estas dos cantidades:

$$C(p,d) = min\left|d'(x_L, x_R, I_L, I_R), d'(x_R, x_L, I_R, I_L)\right|$$

2. **Cálculo de la agregación del coste**:

Una vez calculado el coste $C(p,d)$, para un píxel p con su píxel correspondiente de la imagen contraria $q = e_{bm}(p,d)$ en su linea epipolar, se calcula el coste de agregación o agregación del coste. El cálculo del coste es generalmente ambiguo; correspondencias erróneas pueden tener un coste menor que las correctas debido al ruido, por este motivo, se incluyen restricciones adicionales por penalización en los cambos de disparidades vecinas. El coste y las restricciones se expresan definiendo la energía $E(D)$ que depende de la disparidad D; esta energía tiene en cuenta la suma de todos los costes de disparidad, la penalización P_1 para pequeños cambios en disparidades vecinas y la penalización P_2 para grandes cambios de disparidad; siendo necesario siempre que $P_2 > P_1$

$$E(D) = \sum_{x_R}((C(p, D_p) + \sum_q P_1 T[|D_p - D_q| = 1] + \sum_q P_2 T[|D_p - D_q| > 1]))$$

El problema puede plantearse como la búsqueda del mapa de disparidad D que minimice esta función de energía. El calculo del coste de agregación se realiza en 1D para todas las las direcciones; el coste de agregación $S(p,d)$ para un píxel p con disparidad d se calcula con el sumatorio de todos los costes mínimos en 1D por direcciones que terminan en dicho píxel y dicha disparidad. El coste $L_r(p,d)$ a lo largo de una dirección r del píxel p y con disparidad d se define recursivamente como:

$$L_r(p,d) = C(p,d) + min(L'_r(p-r,d),$$
$$L_r(p-r,d-1) + P_1,$$
$$L_r(p-r,d+1) + P_1,$$
$$min_i L_r(p-r,d-i) + P_2) - min_k L_r(p-r,k))$$

La ecuación añade así los mínimos costes de píxeles previos $p - r$ en la dirección incluyendo las penalizaciones de variación de disparidad $P1$ y $P2$. El número de direcciones consideradas en este cálculo serán cinco; por lo que finalmente, con $r = 5$:

$$S(p,d) = \sum_r L_r(p,d)$$

3. **Cálculo de la disparidad**:

   Finalmente, el valor de disparidad se selecciona mediante el método "Winner takes all", en el que se asigna a cada píxel la disparidad correspondiente al menor coste de agregación.

3.2.1. Parámetros influyentes en el cálculo

Los parámetros influyentes para el cálculo de la disparidad y la generación del mapa de disparidad han sido, en relación a los cálculos realizados:

- **El valor de disparidad mínimo**, es decir, la diferencia necesaria entre píxeles para que se considere que existe disparidad. Este parámetro es muy útil a la hora de eliminar el ruido provocado por los defectos en la calzada.
- **La diferencia entre los valores de disparidad mínimo y máximo**, que define el rango de disparidad de la imagen.
- **El tamaño de bloque sobre el que calcular la disparidad**, cuyo valor sería igual a 1 en caso de querer calcularla sobre cada píxel individual. La importancia del tamaño y dimensiones del área cuya correspondencia se quiere comprobar radica en que la robustez de la correspondencia aumenta con áreas amplias; sin embargo la suposición de que la disparidad es constante en dicha área provoca bordes borrosos cuando se producen discontinuidades de disparidad; esto puede reducirse, pero no evitarse, como puede verse en alguno de los ejemplos sobre disparidad (fig 3.3).

- **Los parámetros de control de la uniformidad** de la disparidad, es decir, los valores de penalización P_1 y P_2 ya mencionados en cambios de disparidad de pequeño valor (diferencias en una unidad) respecto a los píxeles vecinos y en cambios de disparidad mayores respectivamente.
- **La máxima diferencia** permitida entre píxeles correspondientes de las imágenes derecha-izquierda, de modo que no se tomen píxeles parecidos como pares correspondientes.

Con todo esto y tras varias pruebas, se obtiene el mapa de disparidad tal y como se muestra en el ejemplo de la figura (fig 3.3).

3.3. Obtención del uvt-disparity

A partir del mapa de disparidad, se procede a crear las imágenes ut-disparity y vt-disparity, que como ya se ha mencionado en el capítulo sobre el estado del arte, sección 2.1.1, son el resultado de la concatenación de las imágenes u-disparity y v-disparity de hasta cuatro instantes consecutivos; se implementa el cálculo de ambas imágenes con el fin de descubrir cual de ellas resulta mas representativa para la función del sistema propuesto.

El u-disparity se obtiene como una acumulación de los histogramas laterales del mapa de disparidad por columnas, mientras que el v-disparity es una acumulación de los mismos por filas, dando como resultado las imágenes que se pueden ver en la figura 3.4. Como puede observarse en la figura, el u-disparity muestra información de la geometría presente frente al vehículo y la distribución de obstáculos vistos desde el mismo, mientras que el v-disparity muestra la geometría de la calzada, así como la presencia de objetos que, al ser vistos lateralmente puede aportar información sobre la distancia a la que estos se encuentran del vehículo, pero no aporta información sobre la colocación real de los mismos. A primera vista, resulta mas descriptiva la información del u-disparity para el sistema que se quiere implementar; por poner un ejemplo, en el caso de estudio de un peatón cruzando por delante del vehículo, en el v-disparity no se apreciaría el movimiento del mismo al estar posicionado en la misma línea horizontal, mientras que en el u-disparity se podría ver la evolución y movimiento de la disparidad correspondiente al peatón.

Tabla 3.1: Resultados del test para la elección entre u-disparity y v-disparity

Imagen	Área bajo la curva ROC(%)	Área bajo la curva precision-recall (%)	Precisión por validación cruzada (%)
udisparity	0.966	0.972	0.87 (+/- 0.15)
vdisparity	0.929	0.934	0.79 (+/- 0.19)

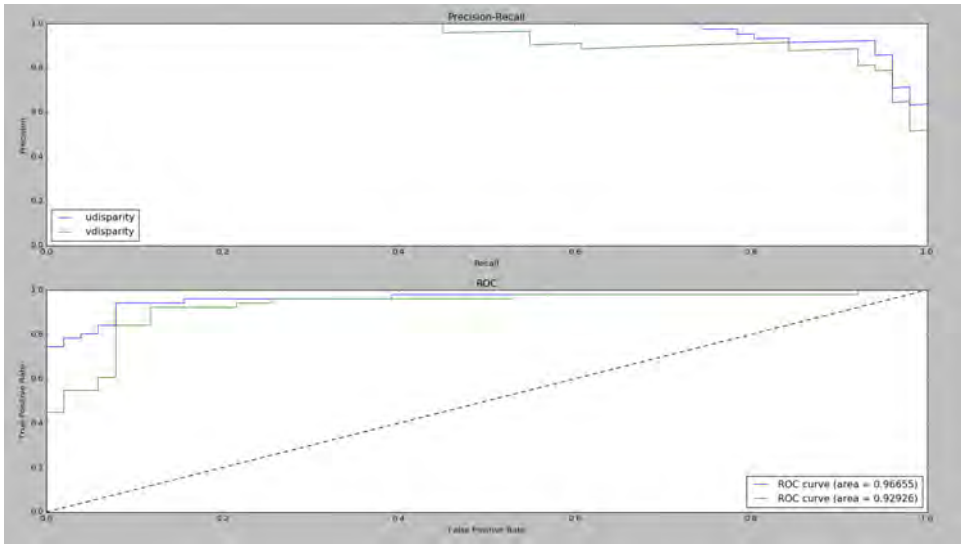
Para comprobar cual de las dos representaciones resulta mas descriptiva, se ha realizado el entrenamiento y posterior test sobre las imágenes ut-disparity y vt-disparity construidas con secuencias de cuatro instantes para cada situación; a la vista de los resultados, cuyas curvas de calidad (definidas en la sección 2.3.1 de este proyecto) pueden verse en la figura 3.5; se comprueba la suposición inicial, es mas eficaz emplear el cálculo del u-disparity, con el cual se obtienen mayores valores de precisión, como puede verse en la tabla 3.1 gracias a los resultados de las áreas bajo las curvas de calidad y de la precisión media por validación cruzada, pues permite un nivel

(a)

(b)

(c)

Figura 3.4: Resultados u-disparity y v-disparity para los casos de estudio: un vehículo delante (a), un peatón cruzando (b) y calzada libre (c) a partir de los mapas de disparidad del apartado anterior



(a)

Figura 3.5: Comparación gráfica de los resultados del clasificador para las imágenes vt-disparity y ut-disparity

de percepción mayor de los objetos y obstáculos que se encuentran delante del vehículo y su posición respecto a este.

3.3.1. Parámetros influyentes en la generación de la imagen ut-disparity

Si bien la creación de la imagen ut-disparity no supone un cálculo de gran complejidad matemática ni con un gran número de parámetros a modificar, si se ha estudiado la influencia tanto del número de instantes tenidos en cuenta como del tratamiento posterior y umbralización de la imagen generada.

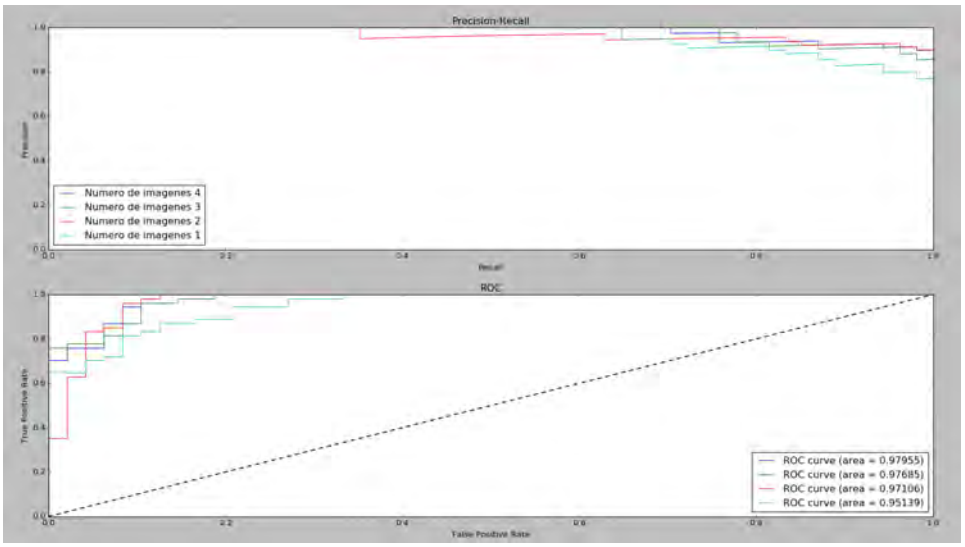
- **Número de imágenes empleadas**

  Una vez decidida la utilización del u-disparity, el parámetro fundamental en la construcción de la imagen ut-disparity ha sido el número de imágenes a representar. En la base de datos escogida, puede obtenerse información de hasta 4 instantes consecutivos, sin embargo esto no quiere decir que deban utilizarse necesariamente estos cuatro instantes, pues como es lógico el tiempo de cómputo será mayor cuanto mayor número de datos se procesen y cuanto mayor sea el tamaño la imagen

final. Por este motivo se ha comprobado la influencia en los resultados de clasificación y predicción del número de imágenes representadas en el ut-disparity desde un único instante hasta los cuatro disponibles (figura 3.6). Se observa en este estudio, como era de esperar, que un mayor número de instantes considerados resulta en un mejor clasificador, con una mayor precisión y parámetros de calidad, como se muestra en la tabla 3.2, siendo mejor el clasificador generado a partir de los cuatro instantes por poseer una mayor cantidad de información; será este el número de secuencias considerada para este trabajo, no obstante, si fuera primordial reducir el tiempo de cómputo del sistema podría escogerse la opción con tres instantes.

Tabla 3.2: Resultados del test para distinto número instantes por situación

Número de instantes	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión por validación cruzada (%)
4	0.979	0.981	0.84 (+/- 0.08)
3	0.977	0.979	0.81 (+/- 0.10)
2	0.971	0.967	0.83 (+/- 0.06)
1	0.951	0.958	0.79 (+/- 0.09)



(a)

Figura 3.6: Comparación gráfica de resultados del clasificador para diferente número de secuencias de imágenes por situación

- **Umbralización de resultados**

Como última fase de este apartado, se realiza una umbralización de la imagen ut-disparity creada con el fin de reducir el ruido y resaltar únicamente los valores e información correspondientes a los obstáculos presentes en la imagen (Lancis, 2014). La umbralización se realiza por tanto para los valores bajos de intensidad de píxel; tras varias pruebas centradas en el resultado final del clasificador, cuyos resultados pueden verse en las curvas de calidad de la figura 3.7 y en los valores de precisión de la tabla 3.4, introducidos en el capítulo sobre el estado del arte y conceptos, sección 2.3.1; se fija este umbral con un valor de 20, siendo esta la mínima intensidad a tener en cuenta y obviando todos los píxeles con una intensidad menor a este valor.

Finalmente y como puede verse en la figura 3.9, con el fin de reducir el tamaño de la imagen se ha eliminado la sección libre de datos del ut-disparity, reduciéndose la cantidad de datos a procesar sin perder ningún tipo de información.

Tabla 3.3: Resultados para los distintos valores de umbralización de la imagen ut-disparity

Valor umbral	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
10	0.964	0.943	0.85 (+/- 0.04)
15	0.964	0.951	0.87 (+/- 0.07)
20	0.968	0.961	0.87 (+/- 0.07)
25	0.948	0.938	0.82 (+/- 0.11)
30	0.933	0.899	0.78 (+/- 0.12)

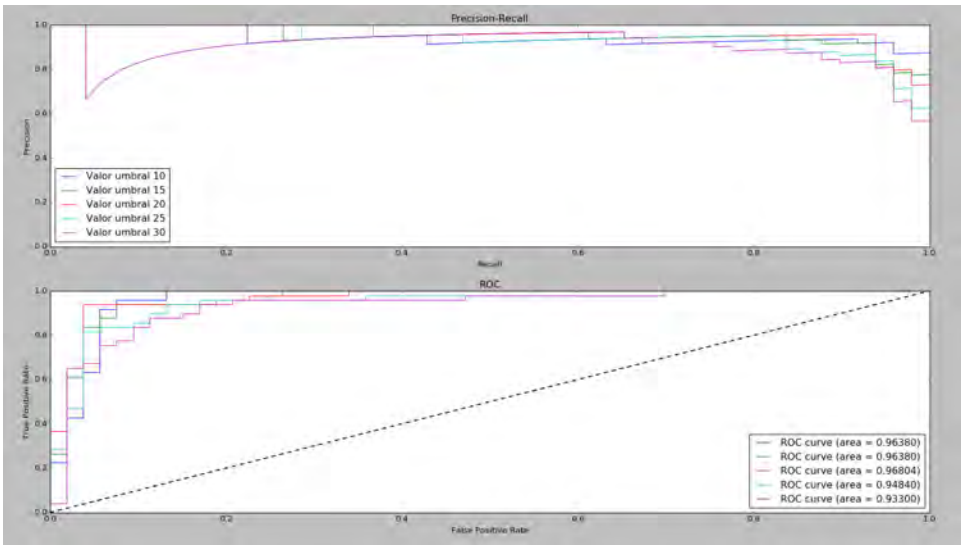


Figura 3.7: Distintos valores de umbralización para el ut-disparity

3.4. Histograma de gradientes orientados (HOG)

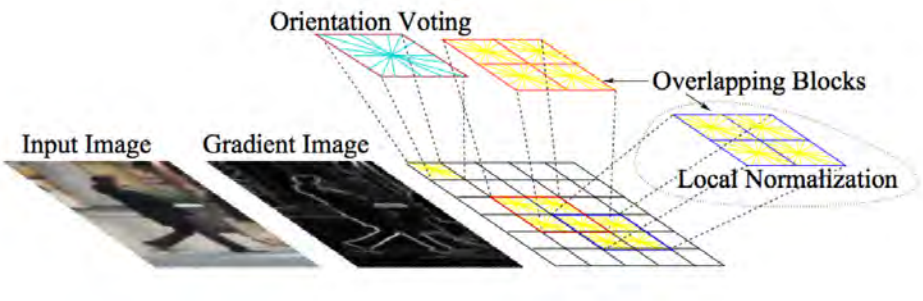


Figura 3.8: Representación esquemática del proceso de cálculo del Histograma de Gradientes Orientados (Dalal y Triggs, 2005)

Una vez obtenida la imagen ut-disparity, se procede a calcular su histograma de gradientes orientados, la función implementada para ello, perteneciente a la librería sckit-image (van der Walt et al., 2014) para el procesamiento de imágenes, se basa en el conocido estudio de (Dalal y Triggs, 2005) para el reconocimiento de personas a partir del HOG, que en primer lugar divide la imagen en pequeñas regiones espaciales llamadas *celdas* y calcula el histograma de direcciones del gradiente o lo que es lo mismo, la orientación de los bordes, en cada una de estas celdas y, además, realiza una normalización del contraste de cada uno de estos histogramas para aumentar la robustez frente a cambios de iluminación o sombras. La segunda parte del método no es tan decisiva en este trabajo, pues va a realizarse sobre la imagen creada ut-disparity, en la que no habrá cambios de iluminación.

Así pues, el algoritmo implementado en este proyecto, como puede verse en la figura 3.8, sigue los pasos descritos a continuación:

1. **Cálculo del gradiente**

   Se calculan los gradientes horizontal y vertical de la imagen con el fin de capturar información sobre su contorno, silueta y textura. El operador gradiente G aplicado sobre una imagen $f(x, y)$ se define por la función:

   $$\Delta f(x, y) = [G_x G_y] = \left[\frac{\partial f}{\partial x} \frac{\partial f}{\partial y}\right]$$

   Este vector gradiente representa la variación máxima de intensidad para cada punto (x,y), siendo la dirección del mismo perpendicular a los

bordes, proporcionando la información deseada.

2. **Cálculo del histograma de gradientes orientados**:

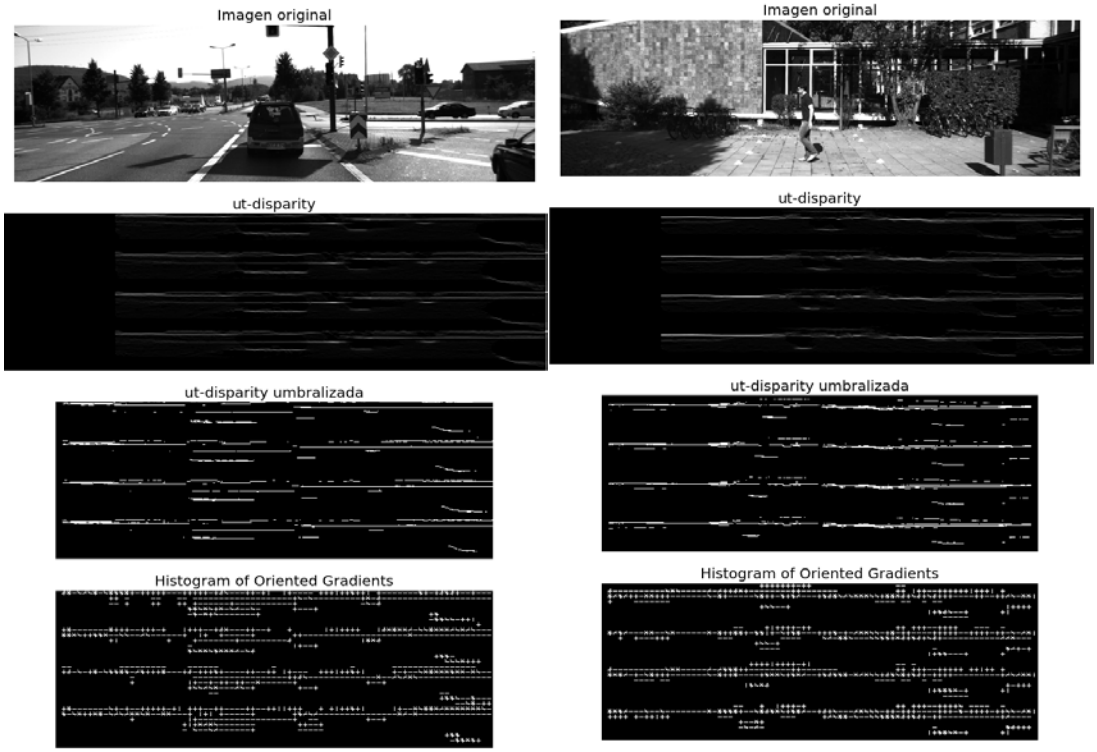
   Se divide la imagen por pequeñas regiones llamadas **celdas** y se va acumulando un histograma de los gradientes orientados local en 1D para todos los píxeles de la misma. Cada uno de estos histogramas divide el rango del ángulo del gradiente en una serie de subrangos, pudiendo estar contenida esta orientación en un rango total de entre 0 y 360 grados y pudiendo especificar la cantidad de subrangos u orientaciones a tener en cuenta.

3. **Normalización por bloques**:

   Se acumulan los histogramas locales de grupos de celdas a los que llama **bloques**, que pueden estar superpuestos entre sí, pudiendo pertenecer una misma celda a varios bloques. Así, al normalizar el contraste de estos bloques, en el vector de salida podrá aparecer la misma celda con distintas normalizaciones. Los bloques se forman horizontalmente recorriendo las columnas de las celdas de izquierda a derecha y verticalmente, recorriendo las filas de las celdas desde arriba hacia abajo. Esta normalización final por bloques es lo que se llama el descriptor HOG.

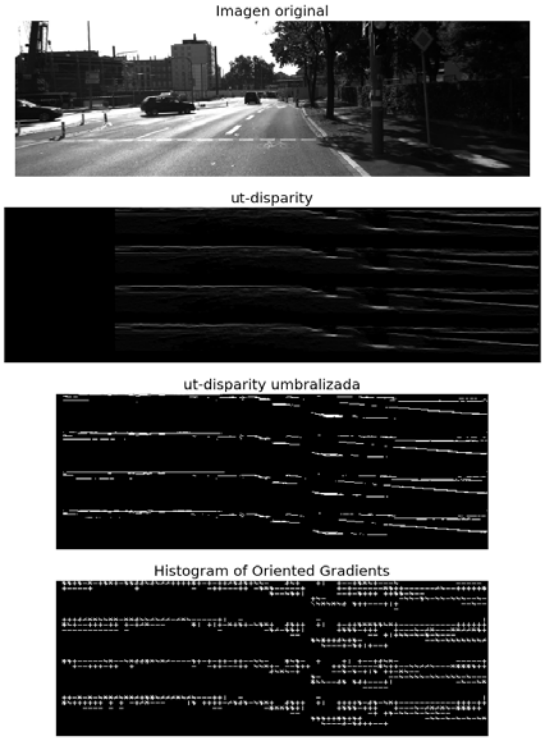
4. **Vector e imagen de salida**:

   Finalmente se colocan los HOG calculados de todos los bloques en un vector unidimensional, que será el utilizado como descriptor para el entrenamiento y se crea, además, una imagen como representación visual del Histograma de Gradientes Orientados, como puede verse en (fig 3.9)



(a)

(b)



(c)

Figura 3.9: Resultados de los ut-disparity original y umbralizado a partir de los mapas de disparidad y del cálculo del HOG en los casos de estudio, un vehículo delante (a) un peatón cruzando (b) y calzada libre (c)

3.4.1. Parámetros influyentes en el cálculo

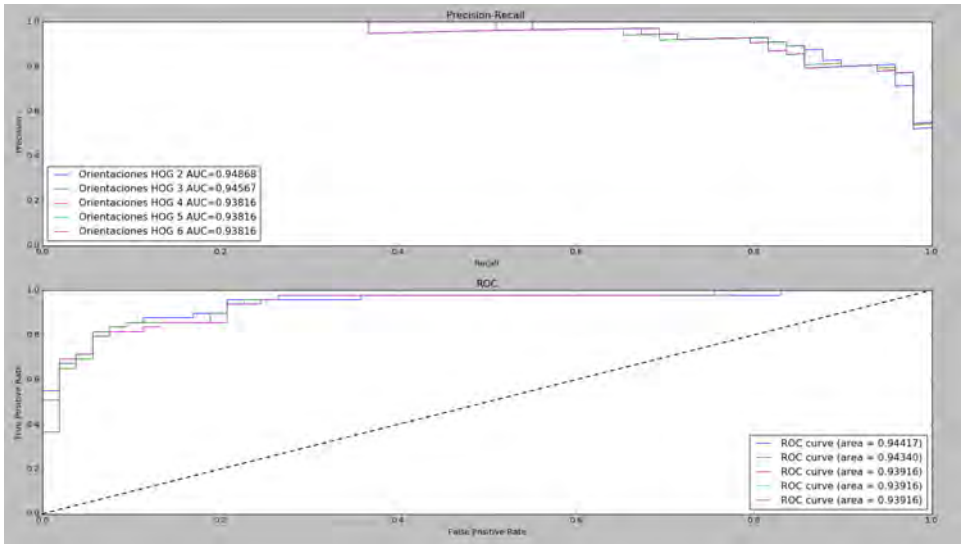
Una vez vistos los pasos que sigue el sistema para el cálculo del HOG, se evalúa la importancia de algunos parámetros influyentes en el mismo, sobre los cuales se ha realizado un estudio comparativo tanto de los resultados obtenidos como del tiempo de cómputo asociado de forma que se puedan escoger para el trabajo final aquellos cuyos resultados son mas óptimos.

- **Número de subrangos de orientación**

  En el cálculo del histograma de gradientes orientados, como ya se ha mencionado, se divide el rango de valores de los ángulos del gradiente en una serie de subrangos; de esta forma se agrupan en el histograma los distintos gradientes de cada píxel en función del subrango al que pertenezcan. Después de diversas pruebas de variación de éstos subrangos, cuyas curvas de calidad pueden verse en la figura 3.10 y cuyos valores de precisión estan incluidos en la tabla 3.4; se llega a la conclusión de que no es necesario un gran número de ellos, obteniéndose buenos resultados con únicamente dos subrangos de orientación; esto tiene sentido debido a la composición de la imagen cuyo HOG se quiere calcular, que no es otra que la imagen ut-disparity generada, una imagen binaria compuesta por líneas en su mayoría horizontales que no requieren de un amplio rango para ser descritas; además, en la práctica, un menor número de subrangos supone un menor tiempo de cómputo.

Tabla 3.4: Resultados para los distintos subrangos de orientación en el cálculo del HOG

Número de subrangos	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
2	0.944	0.949	0.86 (+/- 0.05)
3	0.943	0.946	0.86 (+/- 0.05)
4	0.939	0.938	0.81 (+/- 0.06)
5	0.939	0.938	0.81 (+/- 0.06)
6	0.939	0.938	0.81 (+/- 0.06)



(a)

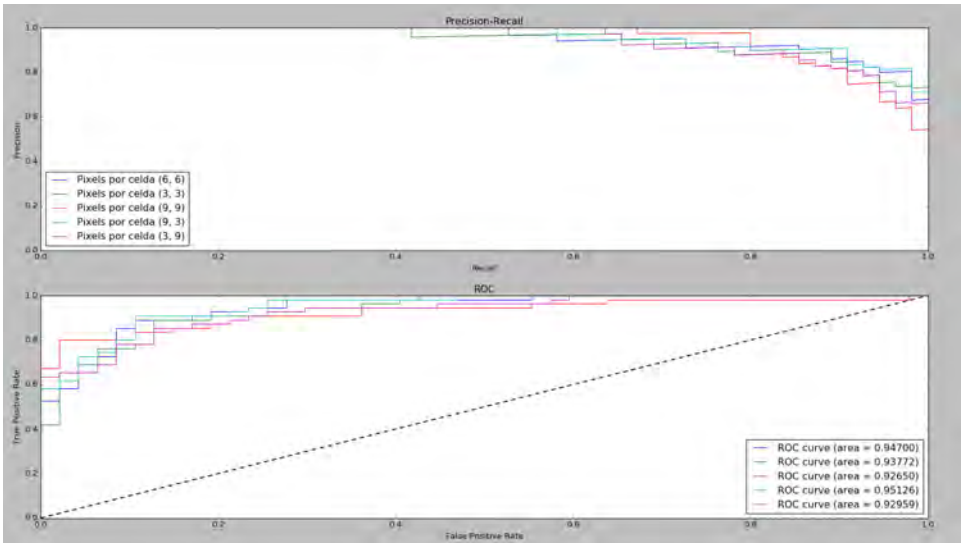
Figura 3.10: Estudio del parámetro de subrangos de orientación para el HOG

- **Número de píxeles por celda**

Las dimensiones de las celdas o grupos de píxeles en los que se calculará el gradiente influye en la cantidad de información a describir, pues si este es demasiado grande, se perderá información de la imagen, y si por el contrario es demasiado pequeño, se pierde la uniformidad en la descripción. Respecto al tiempo de cómputo éste aumenta conforme mas pequeñas son las dimensiones de las celdas, pues el sistema tiene un mayor número de regiones de la imagen en las que calcular el gradiente de orientaciones y por tanto, el vector de salida será también mayor. En las pruebas realizadas sobre este parámetro con distintas dimensiones de celda cuadrada y rectangular, cuyas curvas de calidad se aprecian en la figura 3.11 y cuyos valores de precisión se incluyen en la tabla 3.5, se concluye que para describir adecuadamente la imagen es suficiente con un número de píxeles por celda del orden de $6x6$.

Tabla 3.5: Resultados para los distintos tamaños de píxel por celda en el cálculo del HOG

Tamaño	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
(6 x 6)	0.947	0.955	0.88 (+/- 0.07)
(3 x 3)	0.938	0.946	0.67 (+/- 0.09)
(9 x 9)	0.926	0.951	0.88 (+/- 0.06)
(9 x 3)	0.951	0.959	0.73 (+/- 0.07)
(3 x 9)	0.929	0.946	0.87 (+/- 0.05)



(a)

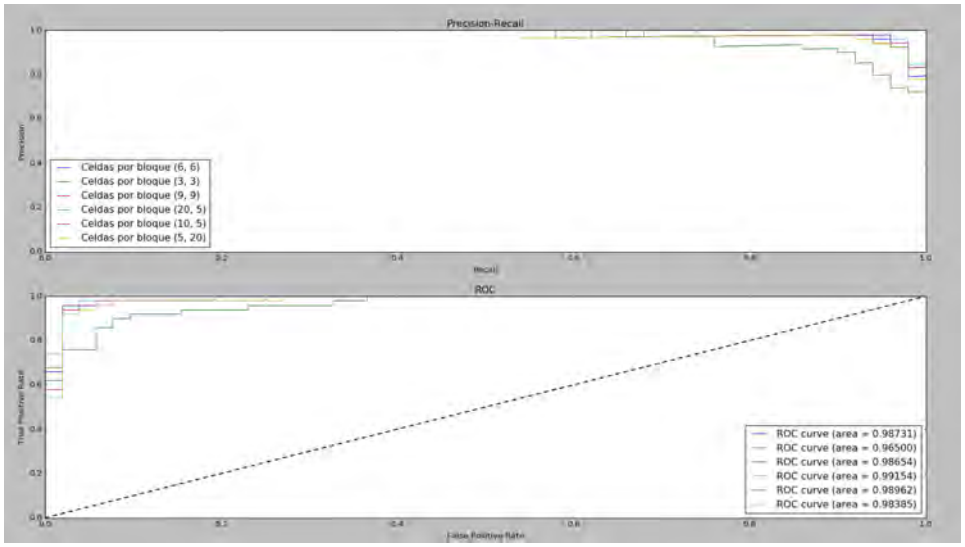
Figura 3.11: Estudio del parámetro de píxeles por celda para el HOG

- **Número de celdas por bloque**

  Por último, para una buena normalización del HOG por grupos de bloques, el número de celdas contenidas en dichos bloques debe ser la adecuada, en este caso y comprobando para diferentes valores de bloques cuadrados y rectangulares, el resultado de dichas comprobaciones puede verse en las curvas de calidad de la figura gráfica 3.12 y en los valores de precisión de la tabla 3.6; a la vista de estos resultados se escoge un valor rectangular de $20x5$ celdas contenidas en cada bloque; en este caso, el número de celdas por bloque no afecta significativamente al tiempo de cómputo

Tabla 3.6: Resultados para los distintos tamaños de celdas por bloque en el cálculo del HOG

Tamaño	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
(6 x 6)	0.987	0.987	0.84 (+/- 0.09)
(3 x 3)	0.965	0.966	0.81 (+/- 0.17)
(9 x 9)	0.986	0.984	0.86 (+/- 0.06)
(20 x 5)	0.991	0.991	0.87 (+/- 0.08)
(10 x 5)	0.990	0.989	0.86 (+/- 0.07)
(5 x 20)	0.984	0.982	0.86 (+/- 0.06)



(a)

Figura 3.12: Estudio del parámetro de número de celdas por bloque para el HOG

3.5. Estandarización del descriptor

Muchos de los estimadores de los métodos de aprendizaje pueden actuar mal si las características individuales de entrada a ellos no tienen el mismo estándar de distribución normal de datos, que sería una gaussiana con media igual a cero y varianza igual a la unidad. Para estandarizar y normalizar los datos, la librería scikit-learn (Pedregosa et al., 2011) posee una serie de funciones que calculan la media y la desviación estándar del conjunto de datos del entrenamiento de modo que pueda aplicarse esta transformación a los posteriores casos a predecir. Las mencionadas funciones de la librería escogida actúan de tres modos diferentes:

- Escalado estándar (*StandardScaler*): estandariza las características mediante la eliminación de la media (centra los datos en cero) y la escala a varianza unidad. Estos dos procesos ser realizan de forma independiente para cada característica y sobre los ejemplos del conjunto de datos destinado al entrenamiento.

- Escalado entre un rango (mínimo, máximo) dado (*MinMaxScaler*)

- Escalado alrededor del máximo absoluto (*MaxAbsScaler*)

Probando cada uno de estos métodos de normalización de los datos, obtenemos los resultados representados en las curvas de calidad de la figura 3.13, con los valores de precisión de la tabla 3.7; donde puede verse que la mejor opción para el caso de clasificación que ocupa este proyecto es la normalización entre el rango mínimo y máximo; se utilizará, por tanto un escalado con rango entre 0 y 1.

Tabla 3.7: Resultados para las distintas normalizaciones del descriptor

Función	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
MinMaxScaler	0.983	0.982	0.91 (+/- 0.11)
StandardScaler	0.955	0.963	0.71 (+/- 0.13)
RobustScaler	0.762	0.744	0.64 (+/- 0.11)

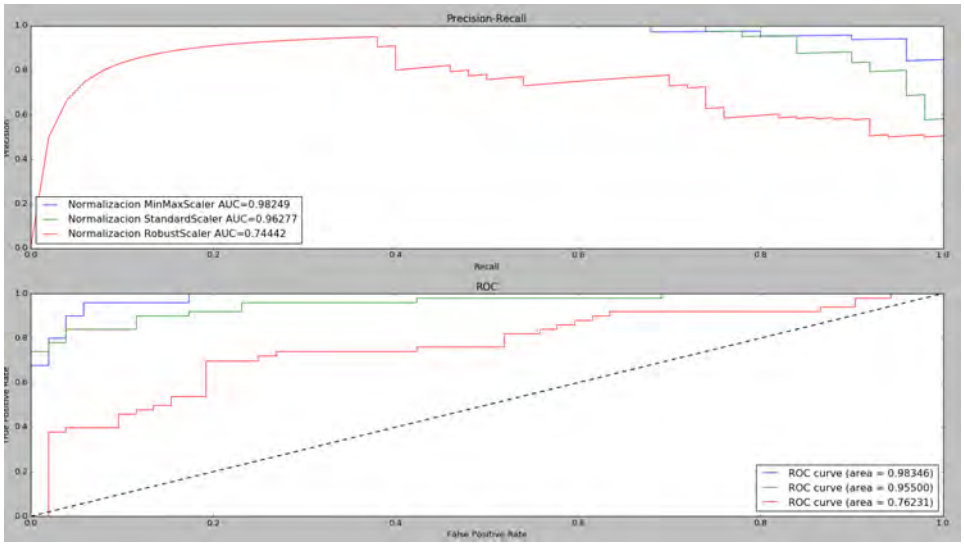


Figura 3.13: Comprobación de distintos métodos de estandarización

3.6. Entrenamiento

Una vez generada la imagen y escogido el descriptor, se procede a realizar el entrenamiento del sistema, para el cual se ha utilizado el método de aprendizaje supervisado de Máquina de Vectores de Soporte (Support Vector Machine, SVM) (Chang y Lin, 2013), ya mencionada en el capítulo 2, sección 2.3, cuyo objetivo es encontrar el hiper-plano que, en el espacio de distribución de los datos, además de separar las distintas clases, maximiza el rango de separación entre ellas. Matemáticamente, SVM pretende optimizar la función cuadrática:

$$min_a J(a) = \alpha Q \alpha^T - e^T \alpha$$

Cumpliendo:

$$0 \leq \alpha_i \geq C$$

$$y^T \alpha = 0$$

Donde e E $\mathbb{R}$ es el vector de unos, C es el parámetro de penalización de error, Q es una matriz de tamaño igual al número de ejemplos a emplear m x m con $Q_{is} = y_i y_s K(x_i x_s)$, donde $K(x_i x_s)$ es la función núcleo, en el caso de este proyecto, una función Kernel y x e y corresponden a los pares de características y clases respectivamente.

En concreto, se ha utilizado la función SVC (Support Vector Classification) de la librería sckit-learn, que permite seleccionar, entre otros, parámetros de penalización de error así como distintas funciones Kernel para distribuciones de datos tanto lineales como no lineales y sus parámetros asociados.

Para generar el estimador mediante un aprendizaje supervisado es necesario separar en primer lugar los datos de los que se dispone en dos particiones, una de ellas para entrenar el clasificador y la otra para realizar las pruebas; el número y variedad de imágenes empleadas para el entrenamiento influye en gran medida en la precisión de las predicciones posteriores, de modo que cuanta mayor información se proporcione en el entrenamiento, mas definida estará la separación entre clases. Para realizar las pruebas en este proyecto se ha escogido una separación de datos de un cuarto del total de imágenes disponibles de cada situación a clasificar, destinándose las tres cuartas partes restantes al entrenamiento; esta separación es necesaria ya que las predicciones no serían fiables si se realizaran sobre las imágenes previamente entrenadas. Del mismo modo, tampoco es fiable realizar las predicciones siempre sobre el mismo grupo de datos, pues esto generaría un sobreajuste de los parámetros del estimador basado en un grupo de entrenamiento y prueba concreto; para evitar esto se recurre a la validación cruzada, un método de

evaluación del clasificador (apartado 2.3.1), que consiste en generar distintas particiones de datos y repetir el proceso de clasificación para cada una de ellas, calculándose finalmente la media aritmética y los errores asociados a cada iteración para estimar la precisión del modelo.

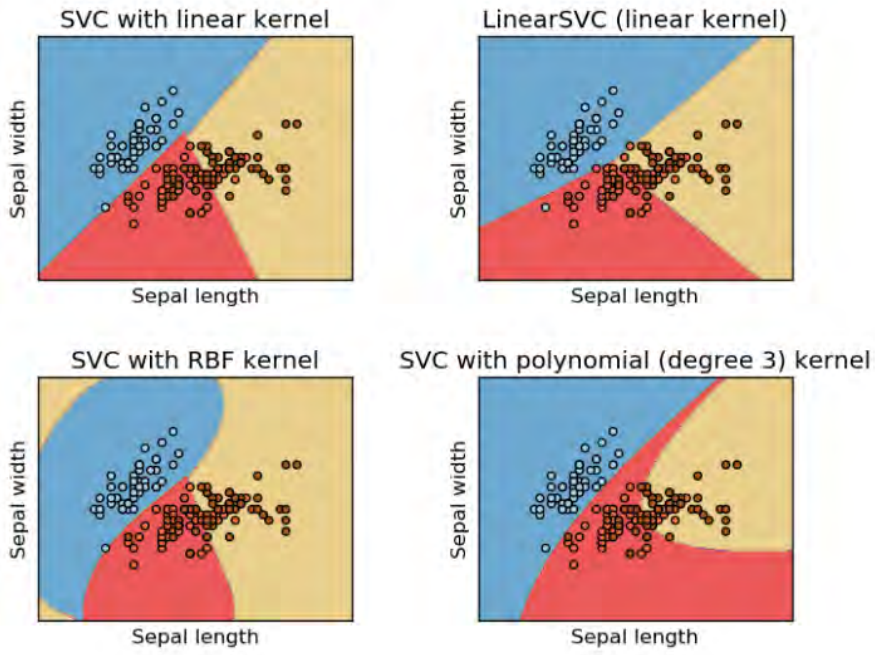


Figura 3.14: Distribución de datos en SVM ((Pedregosa et al., 2011))

3.6.1. Parámetros influyentes en la implementación del clasificador

Los parámetros mas influyentes la hora de generar el clasificador son la tolerancia del mismo, la penalización ante errores y la selección de la función Kernel y sus parámetros asociados.

- **Tolerancia, criterio de parada**

  El criterio de parada marca el número de iteraciones y valor de error máximo que se considera apropiado para una buena implementación del clasificador, marcando la dureza del entrenamiento. Este parámetro es muy importante a la hora de realizar las pruebas de precisión, pues asegura que los resultados de los distintos clasificadores generados

para cada prueba, por ejemplo, las pruebas realizadas para seleccionar los parámetros influyentes en el cálculo del HOG (sección 3.4), sean similares y estén regidas por un mismo criterio. Se escoge, por tanto un valor de este parámetro del orden de 10^{-5} para asegurar la robustez de las pruebas.

- **Penalización del término de error C**

  El parámetro de penalización del término de error, influyente en cualquier clasificador y para cualquier función Kernel, es una relación entre las clasificaciones erróneas de las tuplas de entrenamiento frente a la simplicidad de la superficie de decisión, así, un valor bajo de este parámetro hará la superficie de decisión suave, mientras un valor alto marcará una superficie mas exigente que tratará de clasificar todas las tuplas de entrenamiento correctamente, dando oportunidad al modelo de seleccionar mas ejemplos como vectores de soporte.

- **Función Kernel**

  Como ya se mencionó, en función de la forma de distribución de los datos en el espacio de características, se pueden escoger distintos Kernels para generar el hiper-plano, como son el lineal, el polinómico, el radial (rbf, radial basis function) o el tangente-hiperbólico ($sigmoid$), cuya forma geométrica se puede ver en la representación de la figura 3.14. Una función Kernel $K(x_i, x_s)$, asigna a cada par de objetos de entrada x_i y x_s un valor real que corresponde con el producto escalar de sus respectivas imágenes en el espacio de características, en función del tipo, su formulación matemática es la siguiente:

  Kernel lineal:

  $$K(x_i, x_s) = x_i^T \cdot x_s$$

  Kernel rbf:

  $$K(x_i, x_s) = \exp\left(-\gamma||x_i - x_s||^2\right)$$

  Kernel $sigmoid$:

  $$K(x_i, x_s) = \tanh\left(-\gamma(x_i^T \cdot x_s) + r\right)$$

  Kernel polinómico:

  $$K(x_i, x_s) = \left(\gamma(x_i^T \cdot x_s) + r\right)^d$$

Es de gran importancia que el Kernel escogido se adapte correctamente a la distribución de datos de los casos concretos a clasificar, por ello se realiza un estudio de las distintas funciones y la variación de resultados según los parámetros que se observan en las fórmulas y que definen la forma del hiper-plano:

-Kernel lineal

La distribución lineal de los vectores de soporte es la única que no requiere de ningún parámetro adicional, dando los mismos resultados de precisión para los distintos valores de C comprobados tal y como puede verse en las curvas de calidad de la figura 3.15 y en la tabla de resultados 3.8.

Tabla 3.8: Resultados del test para el Kernel lineal, variación del parámetro C

Parámetros	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
C= 1	0.946	0.963	0.87 (+/- 0.07)
C= 10	0.946	0.963	0.87 (+/- 0.07)
C= 100	0.946	0.963	0.87 (+/- 0.07)
C= 1000	0.946	0.963	0.87 (+/- 0.07)

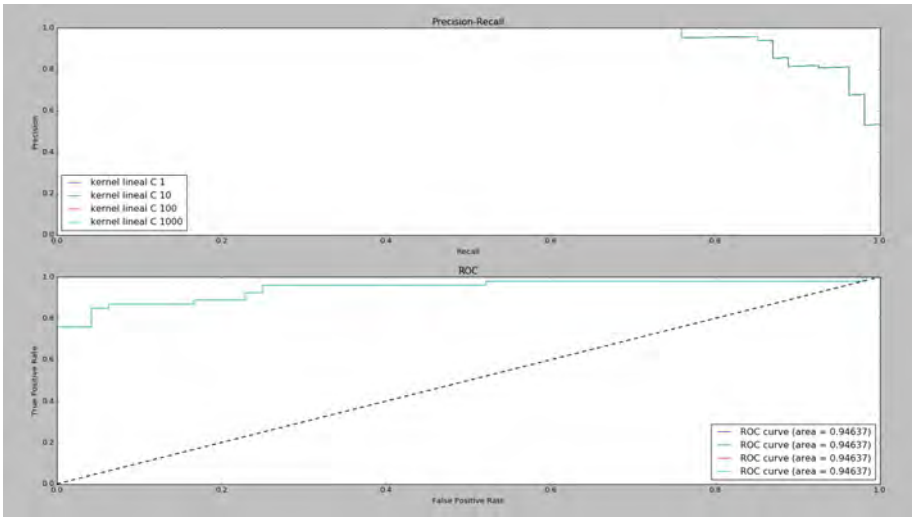


Figura 3.15: Kernel lineal

-Kernel rbf

El único parámetro influyente para la distribución radial es el coeficiente de multiplicación gamma (γ), que define hasta qué punto puede influir en el modelo un solo ejemplo del entrenamiento. Se estudia en este apartado la optimización de este parámetro junto con la influencia del parámetro de penalización de error del clasificador C.

Puede verse en la curva de calidad de la figura 3.16, el estudio del parámetro γ correspondiente a $C = 1$; realizando las pruebas para distintos valores de C se obtienen los resultados detallados en la tabla 3.9, donde se comprueba que el valor de gamma con el cual se obtienen mejores resultados resulta ser del orden de 10^{-4} y $C = 10$ por poseer un mayor área bajo la curva y dar mayores niveles de precisión tanto en pruebas individuales como en la prueba de validación cruzada.

Tabla 3.9: Resultados del test de comparación de los distintos parámetros de la función Kernel rbf; optimización de gamma e influencia del parámetro de penalización de error C

Parámetros	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
C= 1 γ= auto	0.875	0.869	0.51 (+/- 0.01)
C= 1 γ= 10^{-3}	0.934	0.925	0.56 (+/- 0.09)
C= 1 γ= 10^{-4}	0.956	0.953	0.83 (+/- 0.12)
C= 1 γ= 10^{-5}	0.892	0.884	0.70 (+/- 0.17)
C= 10 γ= auto	0.934	0.932	0.75 (+/- 0.13)
C= 10 γ= 10^{-3}	0.935	0.928	0.60 (+/- 0.08)
C= 10 γ= 10^{-4}	0.979	0.977	0.89 (+/- 0.09)
C= 10 γ= 10^{-5}	0.967	0.968	0.87 (+/- 0.06)
C= 100 γ= auto	0.963	0.972	0.86 (+/- 0.13)
C= 100 γ= 10^{-3}	0.935	0.928	0.60 (+/- 0.09)
C= 100 γ= 10^{-4}	0.979	0.976	0.89 (+/- 0.09)
C= 100 γ= 10^{-5}	0.964	0.971	0.86 (+/- 0.12)
C= 1000 γ= auto	0.960	0.970	0.86 (+/- 0.13)
C= 1000 γ= 10^{-3}	0.935	0.928	0.60 (+/- 0.09)
C= 1000 γ= 10^{-4}	0.979	0.977	0.89 (+/- 0.09)
C= 1000 γ= 10^{-5}	0.965	0.971	0.86 (+/- 0.12)

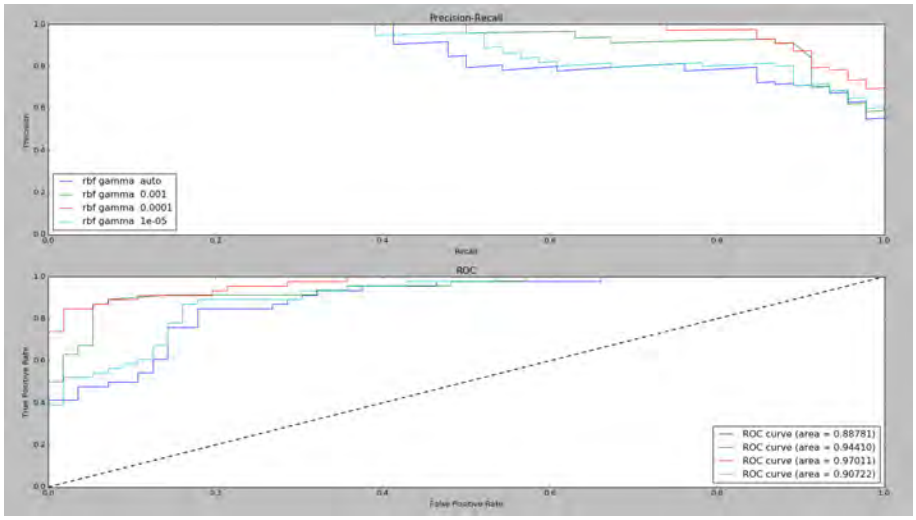


Figura 3.16: Kernel rbf, optimización coeficiente γ

-Kernel sigmoid

Para la función tangente-hiperbólica, se estudian los parámetros gamma (γ) y r en función de C; puede observarse en la figura 3.17), el estudio correspondiente a $C = 1$; y los resultados detallados para distintos valores de C que mejores resultados han reportado pueden verse en la tabla 3.11. En esta tabla, se puede observar que el clasificador mas óptimo de esta clase es aquel que se genera con los parámetros C= 1 γ= 10^{-4} r= 0, por tener la precisión mas alta.

-Kernel polinómico

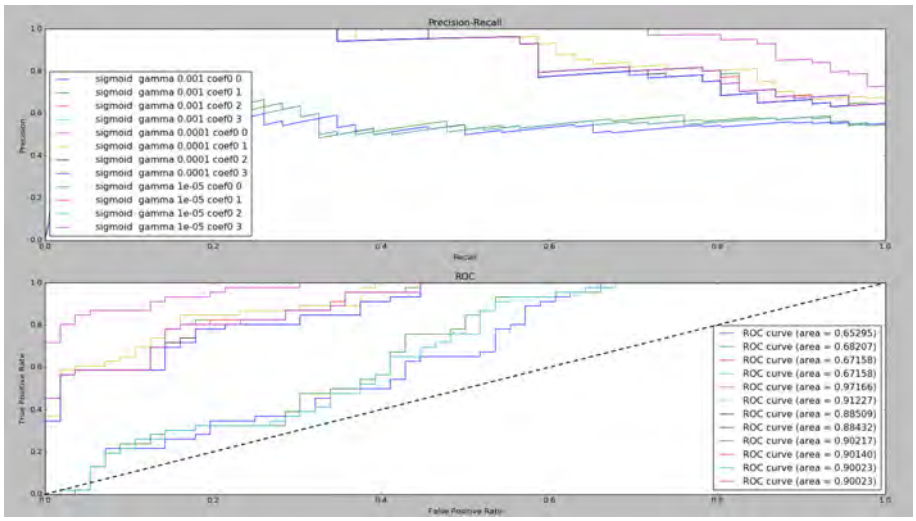
En la función polinómica aparecen mas parámetros a parte del ya mencionado coeficiente gamma (γ), que son el coeficiente de suma coef0 (r) y el grado (degree, d) de la función, que define la flexibilidad del clasificador a la hora de englobar un caso entrante en una clase u otra. Las pruebas de estos parámetros respecto al parámetro C demuestran que los resultados son mejores para un valor de $C = 1$, cuyos valores pueden verse en la figura 3.18; resumiendo los mejores resultados obtenidos en la tabla 3.11 se escoge un factor gamma de $0,01$, un grado 3 y la eliminación del parámetro de suma r (fig 3.18)

Una vez escogidos los mejores valores de las distintas funciones, si se comparan entre sí, se observa que el mejor modo de separación del espacio de

Tabla 3.10: Resultados del test de comparación de los distintos parámetros de la función Kernel sigmoid; optimización de γ y r e influencia del parámetro de penalización de error C

Parámetros	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
C= 1 γ= 10^{-4} r= 0	0.972	0.973	0.80 (+/- 0.08)
C= 1 γ= 10^{-4} r= 1	0.912	0.914	0.72 (+/- 0.07)
C= 1 γ= 10^{-5} r= 0	0.902	0.905	0.73 (+/- 0.08)
C= 10 γ= 10^{-4} r= 0	0.953	0.957	0.87 (+/- 0.05)
C= 10 γ= 10^{-4} r= 1	0.961	0.960	0.83 (+/- 0.04)
C= 10 γ= 10^{-4} r= 2	0.927	0.930	0.73 (+/- 0.07)
C= 10 γ= 10^{-5} r= 0	0.962	0.963	0.86 (+/- 0.10)
C= 10 γ= 10^{-5} r= 1	0.942	0.947	0.75 (+/- 0.11)
C= 100 γ= 10^{-4} r= 0	0.944	0.940	0.76 (+/- 0.19)
C= 100 γ= 10^{-4} r= 2	0.957	0.956	0.83 (+/- 0.07)
C= 100 γ= 10^{-4} r= 3	0.931	0.932	0.76 (+/- 0.07)
C= 100 γ= 10^{-5} r= 0	0.963	0.962	0.89 (+/- 0.03)
C= 100 γ= 10^{-5} r= 1	0.969	0.969	0.88 (+/- 0.04)
C= 100 γ= 10^{-5} r= 2	0.950	0.954	0.78 (+/- 0.09)
C= 100 γ= 10^{-5} r= 3	0.891	0.900	0.74 (+/- 0.12)
C= 1000 γ= 10^{-4} r= 0	0.944	0.940	0.69 (+/- 0.31)
C= 1000 γ= 10^{-4} r= 3	0.951	0.945	0.82 (+/- 0.02)
C= 1000 γ= 10^{-5} r= 0	0.960	0.958	0.89 (+/- 0.03)
C= 1000 γ= 10^{-5} r= 1	0.959	0.958	0.89 (+/- 0.05)
C= 1000 γ= 10^{-5} r= 2	0.965	0.965	0.89 (+/- 0.05)
C= 1000 γ= 10^{-5} r= 3	0.958	0.960	0.84 (+/- 0.11)
C= 10000 γ= 10^{-5} r= 0	0.932	0.945	0.88 (+/- 0.11)
C= 10000 γ= 10^{-5} r= 1	0.930	0.943	0.88 (+/- 0.12)
C= 10000 γ= 10^{-5} r= 2	0.929	0.942	0.88 (+/- 0.12)
C= 10000 γ= 10^{-5} r= 3	0.931	0.944	0.88 (+/- 0.12)

características es el que viene dado por el Kernel polinómico con valores C= 1 γ= 0.01 r= 3 d= 3, esta será la función empleada para crear el clasificador en este proyecto.

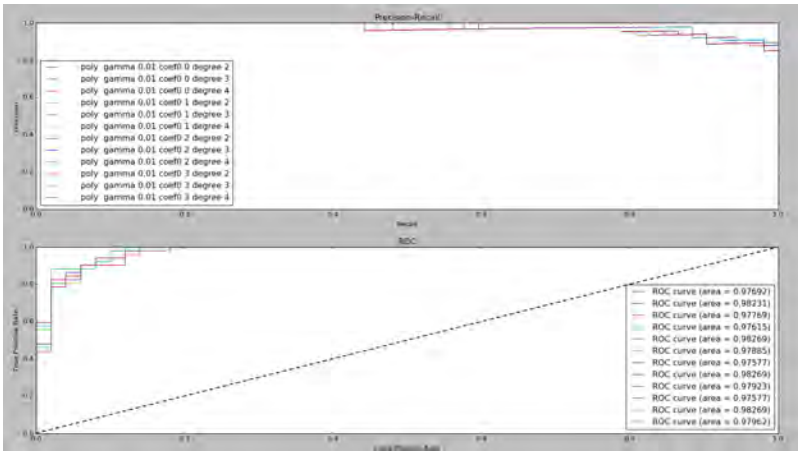


(a)

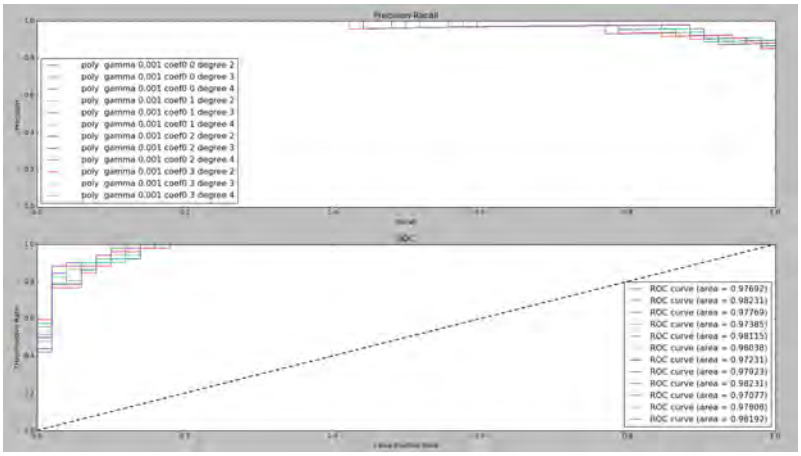
Figura 3.17: Kernel sigmoid, optimización de parámetros r y d

Tabla 3.11: Resultados del test de comparación de los distintos parámetros de la función Kernel polinómica; optimización de γ, r y d e influencia del parámetro de penalización de error C

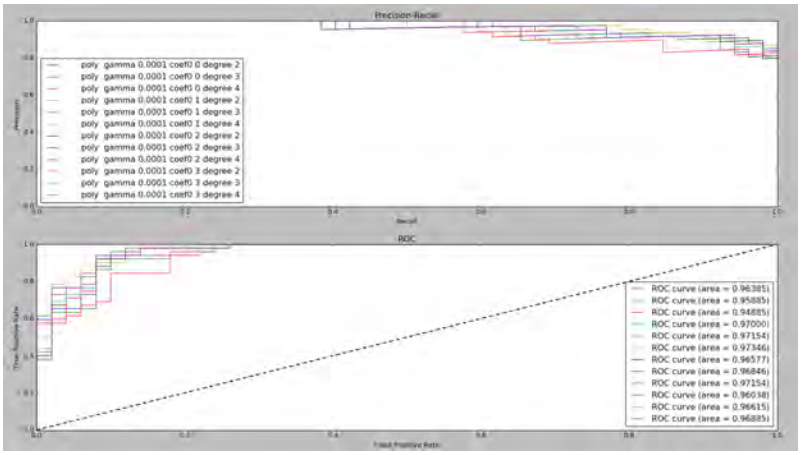
Parámetros	Área bajo la curva (%)	Área bajo la curva precision-recall (%)	Precisión media por validación cruzada (%)
C= 1 γ= 0.01 r= 1 d= 3	0.983	0.986	0.77 (+/- 0.08)
C= 1 γ= 0.01 r= 2 d= 3	0.983	0.983	0.80 (+/- 0.10)
C= 1 γ= 0.01 r= 3 d= 3	0.983	0.983	0.80 (+/- 0.09)
C= 1 γ= 0.001 r= 0 d= 3	0.982	0.984	0.79 (+/- 0.08)
C= 1 γ= 0.001 r= 2 d= 4	0.982	0.984	0.80 (+/- 0.08)
C= 1 γ= 0.001 r= 0 d= 3	0.972	0.973	0.80 (+/- 0.08)
C= 1 γ= 0.001 r= 0 d= 3	0.972	0.973	0.80 (+/- 0.08)



(a)



(b)



(c)

Figura 3.18: Kernel polinómico, optimización de parámetros r, d y γ

3.7. Tiempos de ejecución

Figura 3.19: Características del ordenador empleado

Todo el proceso de cómputo de este proyecto se desarrolla en un ordenador con las características que pueden verse en la figura 3.19. Desde que la cámara situada sobre el vehículo capta el par de imágenes estéreo hasta que se obtiene la clasificación de situación, el sistema pasa por los procesos explicados anteriormente en detalle en este capítulo, siguiendo el diagrama de la figura 3.20; así pues, tiempo que tarda el sistema en dar una respuesta desde que recibe la información depende del tiempo de ejecución de estos algoritmos, cuya temporización media se muestra en la tabla 3.12, dando un tiempo total de algo menos de medio segundo por imagen.

Este trabajo se ha centrado principalmente en optimizar los resultados, para un sistema en tiempo real y con el objeto de minimizar el tiempo de ejecución sería conveniente utilizar otro entorno de programación mas apropiado para ello como la arquitectura CUDA o el lenguaje C/C++.

Tabla 3.12: Temporización

Fase	**Tiempo de cómputo (segundos)**
Mapa de disparidad	0,050313
Imagen u-disparity	0,085859
Concatenación en ut-disparity	<0,0003
HOG	0,185990
Predicción	0,091431
Total	0,4135925674

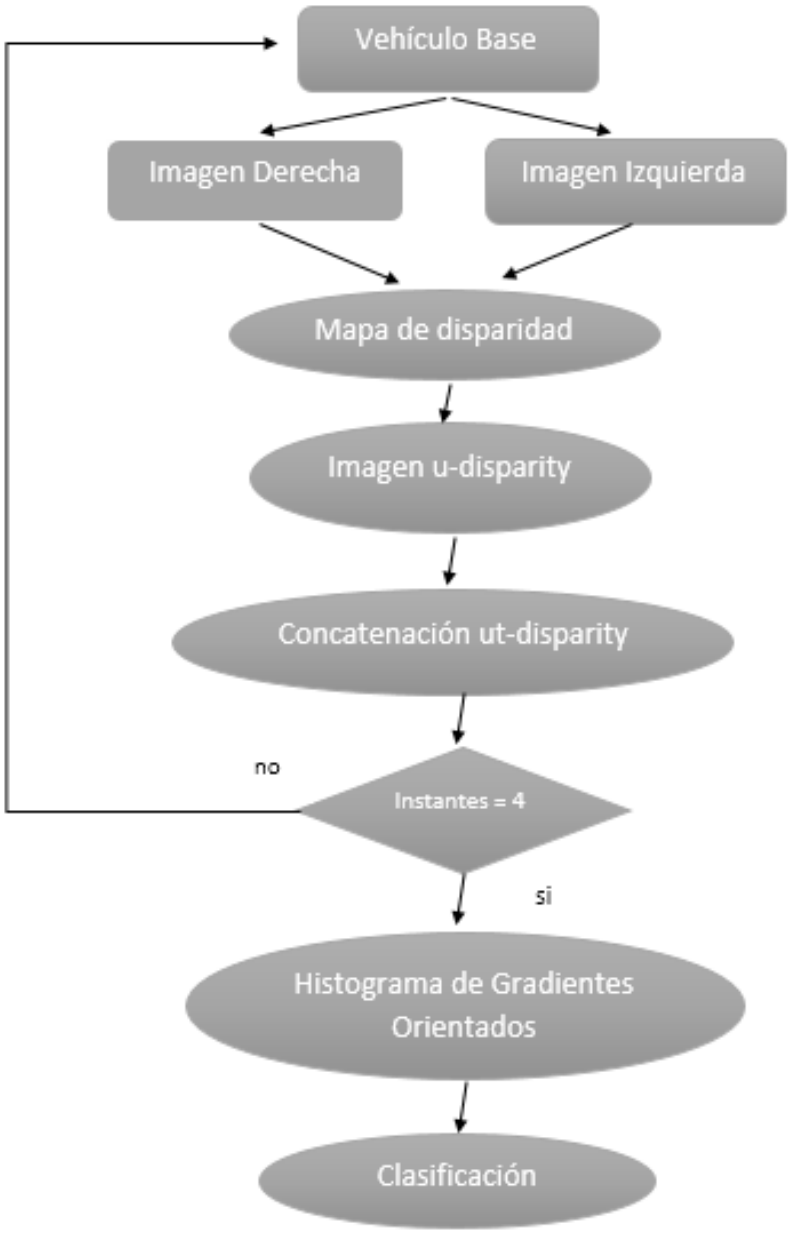


Figura 3.20: Proceso seguido para la detección

Capítulo 4

Resultados

Los resultados obtenidos por el algoritmo propuesto se han comprobado para la clasificación de 3 situaciones en entornos de tráfico tanto urbano como interurbano: un vehículo delante avanzando en la misma dirección y en el mismo carril que el vehículo base (fig 4.1), un peatón cruzando perpendicularmente frente al vehículo (fig 4.2), y la situación en que no ocurre ninguno de estos dos casos (fig 4.3), en esta última se engloban situaciones con ciclistas, trenes, carreteras vacías, peatones circulando en paralelo al vehículo y vehículos que circulan por un carril diferente. En este apartado se detallarán estos resultados obtenidos, analizando las clasificaciones erróneas y sus posibles causas.

(a) (b)

(c)

Figura 4.1: Ejemplos primera situación, vehículo delante

En un principio el caso de estudio de este algoritmo, en el que esta basado la implementación práctica del capítulo anterior, se centró en la clasificación de la primera situación, la de un vehículo circulando en el mismo carril y sentido, sin embargo, mas tarde y con el fin de comprobar en mayor medida

(a) (b)

(c)

Figura 4.2: Ejemplos segunda situación, peatón cruzando

(a) (b)

(c) (d)

(e)

Figura 4.3: Ejemplos tercera situción

la eficacia el clasificador creado se decidió estudiar también la segunda situación, la del peatón cruzando por delante del vehículo; por este motivo el número de imágenes disponibles del primer caso en la base de datos escogida es mucho mayor que las del segundo caso, teniendo un total de 400 ejemplos para el primero y únicamente 50 para el segundo; se estudiará también, por tanto la influencia del número de casos empleados en el entrenamiento.

4.1. Detección de la situación vehículo

En este primer caso, se pretende que el sistema detecte todos los vehículo que circulan por delante y avanzan en el mismo carril que el vehículo móvil en que se sitúan las cámaras; para ello, la base de datos escogida para el proyecto, facilita un total de 400 ejemplos de imágenes con la situación mencionada. Se emplearán, por tanto, estas 400 imágenes mas otras 400 en las que no suceda esta situación; de esta cantidad de imágenes, 1/4 se reservará para el test, utilizándose el resto en el entrenamiento del sistema.

El caso ideal de esta situación es aquel en el que, estando la imagen correctamente iluminada, el vehículo que se encuentra delante circula en línea recta por el carril y su distancia respecto al vehículo base no sea tan elevada como para confundirse con el fondo de la imagen, en este caso, el vehículo aparece representado en la imagen ut-disparity como una linea recta horizontal con apenas desplazamiento en la secuencia, es el caso mostrado en la figura 4.4; sin embargo y como es lógico, en la base de datos escogida aparecen tanto imágenes de este caso concreto ideal como imágenes de casos mas complejos con más obstáculos en los que sigue apareciendo un vehículo delante,

(a)

Figura 4.4: Ejemplo caso ideal vehículo delante

Los resultados de las pruebas para este caso han sido de aproximadamente el 94 % de acierto en la clasificación de las 200 imágenes utilizadas para el test, el 6 % restante correspondiente a clasificaciones erróneas son en su mayoría falsos negativos, es decir, casos en los que la imagen contiene la situación de un vehículo delante pero no se clasifica en el sistema como tal.

Algunas de estas clasificaciones erróneas tienen su explicación por tratarse de vehíuclos que se encuentran muy alejados y no aparecen bien definidos a la hora de construir el ut-disparity, vehículos negros cuya disparidad es mas difícil de calcular, problemas en la iluminación de la imagen ó confusión de determinados objetos, como pueden ser señales de tráfico, con el vehíuclo buscado; algunos de estos casos pueden verse representados en las imágenes de la figura 4.5.

(a)

(b)

(c)

Figura 4.5: Ejemplos detecciones de vehículo fallidas

Con el fin de comparar los resultados de clasificación de los casos vehículo y peatón y teniendo en cuenta el número de imágenes disponible, se ha realizado también un clasificador para el caso vehículo con 51 imágenes, obteniéndose resultados de aproximadamente el 73 % de acierto.

4.2. Detección de la situación peaton

En este segundo caso, el sistema debe detectar todos aquellos peatones que crucen por delante del vehículo; para ello, el número de imágenes del que dispone la base de datos mencionada es bastante menor que en el caso anterior, siendo de un total de 51 ejemplos del caso correcto. Se emplearán, por tanto estas 51 imágenes mas otras 51 en las que no se dé la situación mencionada; de éstas, de nuevo, 1/4 se reserva para el test y el resto se utilizará para entrenar el sistema.

El caso ideal de esta situación es aquel en el que, sin fallos de iluminación en la imagen, se representa un peatón cruzando por delante del vehículo perpendicularmente al mismo, situación en la cual dicho peatón aparecería representado en el ut-disparity como una línea recta cuya posición respecto a la secuencia de imágenes se va desplazando notoriamente en sentido horizontal, tal y como ocurre en la figura 4.6 ; sin embargo tanto para el entrenamiento como para las pruebas se emplean distintas situaciones alejadas de este caso ideal, así encontramos también imágenes con grupos de personas, peatones cuya trayectoria no es del todo perpendicular al vehículo.

(a)

Figura 4.6: Ejemplo caso ideal peatón cruzando

Los resultados de clasificación obtenidos para las imágenes de prueba han obtenido aproximadamente un 75 % de acierto, correspondiendo el porcentaje restante en su mayoría a falsos positivos, es decir, casos en los que no hay ningún peatón cruzando por delante del vehículo y sin embargo se le clasifica como tal, la mayoría de estos errores se deben a que en la imagen aparece un obstáculo de tamaño similar al que pueda tener un peatón, como puede ser un árbol o una motocicleta, y con un movimiento horizontal visto desde el vehículo, tal y como se ve en los ejemplos de la figura 4.7

(a)

(b)

(c)

Figura 4.7: Ejemplos de falsos positivos en la detección de peatón

4.3. Comparación de resultados

Como conclusión de este apartado y a la vista de los resultados para la clasificación de la situación con un vehículo delante, queda reflejado que un mayor número de ejemplos etiquetados para el entrenamiento permite obtener un estimador de mayor precisión en la clasificación, como puede verse en la tabla 4.1.

Tabla 4.1: Resultados de un caso de prueba para diferente número de imágenes disponibles para entrenamiento y test, con una relación de 3/4 y 1/4 del total de ellas para cada función respectivamente

Total de imágenes empleadas	**Resultados (clasificaciones buenas / total de imágenes no entrenadas con las que se ha realizado la prueba)**
800	188/200
102	19/26

Por otra parte y viendo la comparación de resultados para un mismo número de imágenes de entrenamiento, puede observarse que el algoritmo presenta una precisión similar para ambas situaciones, obteniendo entre un 73 % y un 75 % de acierto en las predicciones sobre las imágenes reservadas para el test, como puede observarse en la tabla 4.2.

Tabla 4.2: Resultados de un caso de prueba para la clasificación de las dos situaciones de estudio: un vehículo delante y un peatón cruzando

Situación	**Resultados (clasificaciones buenas / total de imágenes no entrenadas con las que se ha realizado la prueba)**
Vehículo delante	19/26
Peatón cruzando	20/26

Capítulo 5

Conclusiones y trabajos futuros

Una vez descritas y analizadas todas las partes del proyecto, en este último capítulo se procede a exponer las conclusiones generales así como las posibles mejoras y trabajos futuros.

5.1. Conclusiones

Como ya se mencionó en el Estado del Arte, son multitud los estudios sobre sistemas de ayuda a la conducción basados en visión por computador, sin embargo, la mayoría de ellos se centran en segmentar y detectar objetos y obstáculos, siendo escasos los centrados en clasificar situaciones en entornos concretos; el objetivo de este proyecto era por tanto contribuir a aumentar esta rama de investigación y proponer un algoritmo sencillo capaz por si mismo, tras entrenar el correspondiente clasificador, de detectar una situación concreta de tráfico a través de una secuencia de imágenes obtenida del propio vehículo en movimiento.

Trabajar con entornos de tráfico, especialmente con entornos urbanos, supone un gran reto por la multitud de objetos y situaciones que pueden aparecer representadas en la imagen. Para este proyecto en cuestión, no se han realizado distinciones entre entornos; imágenes de autovías, carreteras urbanas, cruces y zonas peatonales, entre otras, son tratadas de la misma forma por el algoritmo presentado, lo que hace especialmente importante la optimización del mismo; por lo que se han realizado los pasos siguientes:

- Elección de los parámetros óptimos del mapa de disparidad; si bien es cierto que en el presente proyecto no se realiza el estudio y comparativa de parámetros, tras comprender y explicar la influencia de cada uno de estos parámetros, se ha escogido una implementación que se conoce

es muy eficaz para las imágenes pertenecientes a la base de datos de la que se han extraido las imágenes.

- Comprobación y justificación del uso de la imagen u-disparity frente a la v-disparity

- Estudio de la influencia del número de secuencias de la situación a clasificar con el fin de no introducir en el sistema información redundante o innecesaria y para crear un clasificador óptimo, con un máximo de 4 instantes disponibles.

- Umbralización y eliminación de ruido y datos innecesarios para el problema descrito una vez construida la imagen ut-disparity.

- Optimización y explicación de los parámetros del descriptor HOG así como la estandarización final del mismo.

- Estudio y comparación de los distintos parámetros para el método de aprendizaje supervisado de máquinas de vectores de soporte así como de las funciones Kernel asociadas a él con el fin de obtener el clasificador mas adecuado.

Finalmente, la eficacia del algoritmo creado se ha comprobado para dos situaciones de clasificación muy diferentes entre si, obteníendose resultados satisfactorios para ambos y demostrándose la capacidad de adaptación del algoritmo a distintas situaciones.

5.2. Trabajos futuros

Respecto a las posibles mejoras para este proyecto, en primer lugar sería interesante ampliar las pruebas del mismo para diferentes situaciones, pues si bien se han realizado pruebas para dos casos de clasificación, los datos de partida sobre la clasificación para el caso de peatones cruzando eran algo escasos, no permitiendo un estudio tan completo y exhaustivo como si que ocurre en el caso de la clasificación de vehículos; además existen muchas otras situaciones que podrían ser testeadas, como situaciones con ciclistas, trenes o adelantamientos laterales de otros vehículos.

También se podría comprobar la respuesta del algoritmo si se utilizase otro descriptor para el ut-disparity, como pueden ser entre otros: el descriptor Blob, que se basa en la detección de puntos brillantes en regiones oscuras de la imagen o de puntos oscuros en regiones brillantes a través de tres posibles algoritmos: la Laplaciana de Gausiana (LoG), la Diferencia de Gausianas (DoG), o el Determinante de Hessian (LoH), siendo el mas preciso el primero y el más rápido el último; el descriptor Gabor, ampliamente utilizado como descriptor de textura; ó el descritpor ODR ((Rublee y Bradski, 2011).

La implementación en tiempo real de este algoritmo podría ser posible si se emplearan Regiones de Interés en lugar de la imagen completa, se modificaran alguno de los parámetros que mas ralentizan el sistema, aunque esto provocara una reducción de la precisión, como pueden ser los parámetros de orientación y número de píxeles por celda correspondientes al cálculo del histograma de gradientes orientados; y empleando entornos de programación mas apropiados como pueden ser la arquitectura CUDA o el lenguaje C/C++.

Por último, para mejorar la eficacia del sistema se podría añadir información adicional a la utilizada en este proyecto, como puede ser la procedente de de sensores adicionales, del aumento del número de imágenes para cada secuencia, de la utilización conjunta de u-disparity y v-disparity en lugar de emplear únicamente uno de ellos, etc, si bien este punto va en contra posición al punto anterior ya que añadiría mas carga computacional al algoritmo y reduciría sus posibilidades de implementación a tiempo real.

Parte II

Apéndices

Apéndice A

Planificación y presupuesto

A.1. Planificación de tareas

Para la realización de este proyecto se ha seguido una planificación de búsqueda de información y desarrollo y optimización del algoritmo conforme a la siguiente tabla, donde se detallan las tareas básicas y críticas así como su secuenciación y temporización estimada.

Fase	Objetivos	Tiempo estimado (horas)
Recopilación de información inicial	▪ Recopilación y comprensión del estado del arte actual sobre temas de visión por computador y, específicamente, sobre detección de acciones y situaciones. ▪ Resumen, explicación y comparación de los estudios actuales sobre ese ámbito.	90

Fase	Objetivos	Tiempo estimado
Búsqueda y adecuación de la base de datos	▪ Búsqueda de una base de datos de imágenes estéreo con las situaciones a identificar por el sistema. ▪ Clasificación e identificación de las situaciones concretas, situaciones con un vehículo delante en el mismo carril y situaciones de un peatón cruzando; y separación de las mismas en casos para el entrenamiento y casos para el test.	30
Generación del mapa de disparidad	▪ Búsqueda y comprensión de la función a implementar y comprobación y optimización de los parámetros de la misma.	40
Generación del uvt-disparity	▪ Cálculo del u-disparity y v-disparity para cada una de las imágenes y concatenación de las mismas para generar el ut-disparity y el vt-disparity. ▪ Comprobación y optimización de resultados en función de varios parámetros.	40
Generación del Histograma de gradientes orientados (HOG)	▪ Búsqueda y comprensión de la función a implementar y comprobación y optimización de los parámetros de la misma.	40

Fase	Objetivos	Tiempo estimado
Entrenamiento del sistema	▪ Generación del código de etiquetado y clasificación para los casos de entrenamiento. ▪ Búsqueda y comprensión de la función a implementar. ▪ Implementación de métodos de comprobación de la calidad del clasificador para la optimización de los parámetros.	80
Documentación del proyecto y exposición de resultados	▪ Documentación de todos los pasos seguidos, así como sus fundamentos teóricos. ▪ Exposición y comparativa de resultados en función de los distintos parámetros modificables en cada fase.	100
		Total: 420 Horas

Tabla A.1: Tabla de planificación de tareas

A.2. Presupuesto

Se pueden dividir los recursos empleados para la realización de este proyecto en dos tipos: recursos humanos y recursos materiales y de software. El primero sería el coste derivado del trabajo de un ingeniero junior que realice todas las tareas de búsqueda, programación y documentación del presente proyecto, por lo que su número de horas asignado será el correspondiente al total del tiempo estimado en el anterior apartado sobre planificación de tareas (A.1). El coste de recursos materiales y de software, por otro lado, serán las herramientas necesarias para que el ingeniero realice el trabajo correctamente, comprendiendo tanto el ordenador utilizado como los programas y software instalado en el mismo.

En la siguiente tabla se describen estos recursos, así como su cantidad correspondiente y el desglose de precios unitarios y totales:

Unidad	Descripción	Medición	Precio unitario (€/unidad)	Precio total (€)
Recursos humanos				
Horas	Salario ingeniero junior	420	10	4200
Recursos materiales y soaftware				
Unidades	Ordenador portátil ASUS: ▪ Procesador Intel Core i7 ▪ Memoria 7,7GiB	1	800	800
-	Sistema operativo ubuntu 14.04 LTS 64bits		Distribución libre	0
-	Gestor de paquetes Anaconda, incluye: ▪ Python 2.7 ▪ Editor de código Spyder	-	Distribución libre	0
				Total: 5.000 €

Tabla A.2: Presupuesto estimado

Bibliografía

Alvarez, J. M., Gevers, T., Lecun, Y. y Lopez, A. M. LNCS 7578 - Road Scene Segmentation from a Single Image. páginas 376–389, 2012.

Birchfield, S. y Tomasi, C. A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling. vol. 20(4), páginas 401–406, 1998.

Bradski, G. Open source computer vision library. *Dr. Dobb's Journal of Software Tools*, 2000.

Chang, C.-c. y Lin, C.-j. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, páginas 1–39, 2013.

Dagli, I., Brost, M. y Breuel, G. Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks. 2002.

Dalal, N. y Triggs, B. Histograms of oriented gradients for human detection. En *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognitio(CVPR'05)*, vol. 1, páginas 886–893 vol. 1. 2005. ISSN 1063-6919.

DGT. Revista dgt, tráfico y seguridad vial 230. `http://www.dgt.es/revista/num230/?pageIndex=18#p=20`, 2015. (Consultada en 18/9/2016).

Enzweiler, M. y Gavrila, D. M. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31(12), páginas 2179–2195, 2009. ISSN 01628828.

Ess, A., Mueller, T., Grabner, H. y Gool, L. V. Segmentation-Based Urban Traffic Scene Understanding. *Procedings of the British Machine Vision Conference 2009*, páginas 84.1–84.11, 2009. ISSN 03010104.

Fitsa. El valor de la seguridad vial. Conocer los costes de los accidentes de tráfico para invertir más en su prevención. páginas 1–25, 2008.

GEIGER, A., LAUER, M., WOJEK, C., STILLER, C. y URTASUN, R. 3D traffic scene understanding from movable platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36(5), páginas 1012–1025, 2014. ISSN 01628828.

GEIGER, A., LENZ, P. y URTASUN, R. Are we ready for autonomous driving? the KITTI vision benchmark suite. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 3354–3361, 2012. ISSN 10636919.

GRAEFE, V. Visual Recognition of Traffic Situations. *Sicici*, (February), páginas 4–9, 1992.

HAAG, M. y NAGEL, H. H. Incremental recognition of traffic situations from video image sequences. *Image and Vision Computing*, vol. 18(2), páginas 137–153, 2000. ISSN 02628856.

HIRSCHMU, H. Stereo Processing by Semiglobal Matching and Mutual Information. vol. 30(2), páginas 328–341, 2008.

HU, Z. y UCHIMURA, K. U-V-disparity: An efficient algorithm for stereovision based scene analysis. *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2005, páginas 48–54, 2005.

KANG, Y., YAMAGUCHI, K., NAITO, T. y NINOMIYA, Y. Multiband Image Segmentation and Object Recognition for Understanding Road Scenes. vol. 12(4), páginas 1423–1433, 2011.

KITTI. The kitti vision benchmark suite. `http://www.cvlibs.net/datasets/kitti/setup.php`, 2016. (Consultada en 09/18/2016).

KLASER, A., MARSZALEK, M. y SCHMID, C. A Spatio-Temporal Descriptor Based on 3D-Gradients. *Proceedings of the British Machine Conference*, páginas 99.1–99.10, 2008.

LABAYRADE, R., AUBERT, D. y TAREL, J.-P. Real time obstacle detection in stereovision on non flat road geometry through"v-disparityrepresentation. vol. 2, páginas 646–651, 2002.

LANCIS, B. M. Análisis de entornos urbanos de tráfico y estimación del movimiento del vehículo para el desarrollo de sistemas avanzados de ayuda a la conducción. 2014.

LAZEBNIK, S. y SCHMID, C. Beyond Bags of Features : Spatial Pyramid Matching for Recognizing Natural Scene Categories. 2006.

LECUMBERRY, F. Cálculo de disparidad en imágenes estéreo, una comparación. *XI Congreso Argentino de Ciencias de la ...*, (September), 2005.

Llorca, D. F., Sotelo, M. A., Hellín, A. M., Orellana, A., Gavilán, M., Daza, I. G. y Lorente, A. G. Stereo regions-of-interest selection for pedestrian protection: A survey. *Transportation Research Part C: Emerging Technologies*, vol. 25, páginas 226–237, 2012. ISSN 0968090X.

Marszalek Marcin. Laptev Ivan, S. C. Actions in Context. (i), páginas 2929–2936, 2009.

Meyer-Delius, D., Sturm, J. y Burgard, W. Regression-based online situation recognition for vehicular traffic scenarios. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, páginas 1711–1716, 2009. ISSN 10504729.

Montes, M. C. Support Vector Machine; Graficas, estadística y minería de datos con Python, Centro de investigaciones Energéticas Medioambientales y Tecnológicas, Madrid. (4), páginas 1113–1123, 2015. ISSN 03875806.

Murray, D. y Little, J. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, vol. 8, páginas 161–171, 2000. ISSN 09295593.

Navneet Dalal, B. T. y Schmid, C. LNCS 3952 - Human Detection Using Oriented Histograms of Flow and Appearance. páginas 1–14, 2006.

NCAP, E. El sitio oficial del programa europeo de evaluación de automóviles nuevos. `http://www.euroncap.com/es`, x. (Consultada en 18/9/2016).

Nedevschi, S., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C., Schmidt, R. y Graf, T. High accuracy stereo vision system for far distance obstacle detection. *IEEE Intelligent Vehicles Symposium, 2004*, páginas 292–297, 2004.

OMS. Las 10 causas principales de defunción en el mundo, `http://www.who.int/mediacentre/factsheets/fs310/es/`. 2012. (Consultada en 18/9/2016).

OMS. Número de víctimas mortales por accidente desde 1960. `http://www.dgt.es/es/prensa/notas-de-prensa/2016/20160104-nuevo-minimo-historico-numero-victimas-mortales-accidente-desde-1960.shtml`, 2016. (Consultada en 18/9/2016).

Oneata, D., Verbeek, J. y Schmid, C. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. *2013 IEEE International Conference on Computer Vision*, páginas 1817–1824, 2013. ISSN 1550-5499.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M. y DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, páginas 2825–2830, 2011.

PERROLLAZ, M., YODER, J. D., SPALANZANI, A. y LAUGIER, C. Using the disparity space to compute occupancy grids from stereo-vision. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, páginas 2721–2726, 2010. ISSN 2153-0858.

PONSA, D., LÓPEZ, A., LUMBRERAS, F., SERRAT, J. y GRAF, T. 3D Vehicle sensor based on monocular vision. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2005, páginas 1096–1101, 2005.

ROYER, E., LHUILLIER, M., DHOME, M. y LAVEST, J.-M. Monocular Vision for Mobile Robot Localization and Autonomous Navigation. *International Journal of Computer Vision*, vol. 74(3), páginas 237–260, 2007. ISSN 0920-5691.

RUBLEE, E. y BRADSKI, G. ORB - an efficient alternative to SIFT or SURF. 2011. ISSN 1550-5499.

SAPIENZA, M., CUZZOLIN, F. y TORR, P. H. S. Learning discriminative space-time action parts from weakly labelled videos. *International Journal of Computer Vision*, vol. 110(1), páginas 30–47, 2014. ISSN 15731405.

SCHARSTEIN, D. y SZELISKI, R. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, vol. 47(1-3), páginas 7–42, 2002. ISSN 09205691.

TEUTSCH, M., HEGER, T., SCHAMM, T. y ZÖLLNER, J. M. 3D-segmentation of traffic environments with U/V-disparity supported by radar-given masterpoints. *IEEE Intelligent Vehicles Symposium, Proceedings*, páginas 787–792, 2010. ISSN 19310587.

WALLACH, H. M. Topic Modeling: Beyond Bag-of-Words Hanna. *Icml2006*, (1), páginas 977–984, 2006.

VAN DER WALT, S., SCHÖNBERGER, J. L., NUNEZ-IGLESIAS, J., BOULOGNE, F., WARNER, J. D., YAGER, N., GOUILLART, E., YU, T. y THE SCIKIT-IMAGE CONTRIBUTORS. scikit-image: image processing in Python. *PeerJ*, vol. 2, página e453, 2014. ISSN 2167-8359.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I. y Schmid, C. Evaluation of local spatio-temporal features for action recognition. *BMVC 2009 - British Machine Vision Conference*, páginas 124.1–124.11, 2009. ISSN 1939-3539.